Consistent circuits for indefinite causal order

Augustin Vanrietvelde[1,2,3], Nick Ormrod*[1,3], Hlér Kristjánsson*[1,3], and Jonathan Barrett[1]

[1]Quantum Group, Department of Computer Science, University of Oxford
[2]Department of Physics, Imperial College London
[3]HKU-Oxford Joint Laboratory for Quantum Information and Computation

arXiv:2206.10042v3 [quant-ph] 26 Nov 2025

Over the past decade, a number of quantum processes have been proposed which are logically consistent, yet feature a cyclic causal structure. However, there is no general formal method to construct a process with an exotic causal structure in a way that ensures, and makes clear why, it is consistent. Here we provide such a method, given by an extended circuit formalism. This only requires directed graphs endowed with Boolean matrices, which encode basic constraints on operations. Our framework (a) defines a set of elementary rules for checking the validity of any such graph, (b) provides a way of constructing consistent processes as a circuit from valid graphs, and (c) yields an intuitive interpretation of the causal relations within a process and an explanation of why they do not lead to inconsistencies. We display how several standard examples of exotic processes, including ones that violate causal inequalities, are among the class of processes that can be generated in this way. We conjecture that this class in fact includes all unitarily extendible processes.

Contents

1 Introduction **2**

2 Reconstructing the quantum switch **5**
2.1 Supermaps and the quantum switch . . . 5
2.2 Routed quantum circuits . . . 7
2.3 Extracting the relevant structure: routed circuit decomposition, skeletal supermap, routed graph . . . 9
2.4 Checking for validity . . . 11
2.4.1 Bifurcation choices and bi-univocality . . . 13
2.4.2 The branch graph and weak loops . . . 14
2.5 Why do we need bi-univocality? . . . 16

Augustin Vanrietvelde: vanrietvelde@telecom-paris.fr,
Current address: Télécom Paris, Palaiseau

Nick Ormrod*: normrod@perimeterinstitute.ca,
Current address: Perimeter Institute for Theoretical Physics, Waterloo

Hlér Kristjánsson*: hler.kristjansson@umontreal.ca,
Current address: Department of Computer Science and Operations Research, Université de Montréal

Jonathan Barrett: jonathan.barrett@cs.ox.ac.uk

*These two authors made equal contributions to this work.

3 The framework **18**

4 Examples of constructing processes with indefinite causal order **24**
4.1 The quantum 3-switch . . . 25
4.1.1 The routed graph . . . 25
4.1.2 The routed circuit . . . 29
4.2 The Grenoble process . . . 30
4.2.1 The routed graph . . . 31
4.2.2 The routed circuit . . . 35
4.3 The Lugano process . . . 38
4.3.1 The logical structure . . . 39
4.3.2 The routed graph . . . 40
4.3.3 The routed circuit . . . 42

5 Discussion and outlook **44**

A The relationship between the supermap and process matrix representations **48**
A.1 At the conceptual level . . . 49
A.2 At the mathematical level . . . 50

B Technical definitions and proofs **53**
B.1 Notations . . . 53
B.2 Technical definitions on supermaps . . . 54
B.3 Technical presentation of the framework . . . 54
B.4 Proof . . . 57
B.4.1 Preliminary lemmas and definitions . . . 57
B.4.2 The induction hypothesis . . . 61
B.4.3 Proof of the base case . . . 64
B.4.4 Proof of the induction step . . . 74

1 Introduction

As traditionally understood, quantum theory is *compositional*. On the one hand, one can apply two channels, one after the other, resulting in an overall process which is itself a quantum channel. On the other hand, two quantum channels can be applied at the same time, each to a different system, defining a quantum channel on the bipartite system. In other words, two quantum channels can be *composed*, in sequence or in parallel, to form another quantum channel. This notion of compositionality is generalized by the quantum circuit formalism [1, 2, 3, 4, 5], in which arbitrary quantum channels can composed by 'wiring them up' into a circuit, and any circuit defines a valid quantum channel.

All of this is very convenient, because compositionality turns out to be a very useful and illuminating feature of the theory. Most obviously, compositionality provides a convenient way of discovering new quantum channels: begin with a set of known quantum channels, and compose them together to form a circuit of your choosing. Once this is done, there is no need to check that the circuit defines a valid quantum channel; the circuit construction itself is a proof that it does. It follows that circuit constructions can be used not only to discover new channels, but also to verify that a given process is in fact a valid quantum channel.

A circuit can be 'tweaked' by replacing the individual quantum channels in the circuit without changing the way in which they are composed to form a circuit. Given a circuit

construction of a channel, a vast class of related processes is obtained by tweaking, and every one of these process must be a quantum channel. Thus a circuit construction provides a *compositional explanation* of why a process is a valid quantum channel: it is valid not because of any specific facts about the individual channels in the circuit, but solely as a result of the way in which they are composed.

Thus compositionality helps one discover, verify, and explain quantum channels. In addition, the circuit formalism facilitates the development of graphical calculi that help speed up calculations, prove theorems, and better appreciate symmetries of processes that are suppressed by more conventional, one-dimensional notation [5, 6].

However, in the last two decades, a class of quantum processes have been proposed that call into question whether compositionality is a completely general feature of quantum theory [7, 8, 9, 10, 11, 12, 13]. These *higher-order* processes act not on states, but on quantum channels. They are typically represented by supermaps [8, 9], or, equivalently, process matrices [10]. Many of them place their input channels in an 'indefinite causal order', meaning that they cannot be constructed as a circuit without *feedback loops*. Feedback loops — forbidden by the standard circuit formalism — are formed when a wire coming out of the circuit is plugged into a wire going into the circuit. They are problematic because when quantum channels are wired together to form a 'loopy circuit' it is *not* guaranteed that the circuit itself defines a quantum channel. Introducing loops into the circuit formalism means sacrificing one of its most appealing features: that a circuit construction is a proof of the validity of the constructed process.

Without this crucial feature, the circuit framework loses much of its appeal. Circuit construction is no longer a safe way of discovering new processes, since a loopy circuit of channels might fail to define a valid process. A circuit that does define a valid process cannot be safely tweaked, and thus does not provide a compositional explanation of the process it defines. It is doubtful that the loopy circuit formalism can facilitate the development of useful graphical calculi.

Since a circuit construction of a process with indefinite causal order is not a proof of its consistency, one needs other proof methods. Existing methods for demonstrating the consistency of a process often involve brute-force, numerical analysis on its *process matrix* [10]. Such methods become impractical as the numbers of dimensions and parties increase. Even when these methods work, they provide little or no conceptual insight.

We are therefore forced to confront the question of whether compositionality is a general feature of quantum theory, or a notion that applies only to those quantum processes that lack indefinite causal order. Should we infer from the issues with feedback loops that processes with indefinite causal order are not compositional? Or do we simply lack the right formalism to see the compositional properties that these processes have? The stakes are high, since our answer to this question will determine whether the many advantages that compositionality brings to quantum processes with definite causal order carry to the indefinite case.

This paper shows that the notion of compositionality in quantum theory generalizes beyond processes with definite causal order. The main result is a theorem showing that whenever a circuit construction follows certian rules, it defines a valid process, even if the construction uses feedback loops and the circuit has indefinite causal order. This leads to a refined notion of compositionality, according to which processes can be composed only when these conditions are met. A circuit formalism based on this notion of compositionality recovers the key feature that is lost when loops are added to standard quantum circuits: that a circuit construction of a processes is a proof of its validity. As a result, other benefits of compositionality mentioned above are also restored, including the possi-

bility of 'tweaking' processes and providing compositional explanations of the validity of a process.

While we do not show that all higher-order processes can be constructed using the framework, we do construct a number of prominent examples in the literature, including the quantum switch [8, 9] and 3-switch [14], the recently proposed Grenoble process [13], and the Lugano process (also called Baumeler-Wolf or Araújo-Feix) [11, 12, 15]. We are led to conjecture that all 'unitarily extendible' [15] processes can be constructed using the framework. We thus show that many significant processes with indefinite causal order admit a compositional explanation, and conjecture that all unitarily extendible processes do too.

Let us now describe our circuit framework in a little more detail. The circuits in the framework provide more fine-grained information about the causal structure than is available using standard circuits with loops. They are based on an extension to the standard quantum circuit framework known as routed quantum circuits, introduced in Ref. [16] (see also Refs. [17, 18, 19, 20]), in which not only tensor product but also direct sum structures are graphically represented. This matters for us because our rules of compositionality refer to those direct sum structures.

The rules are articulated with the help of a *routed graph*, which provides an abstract description of how processes are to be composed. Not all routed graphs are valid, in the sense that when one composes quantum channels in accordance with the routed graph, the resulting circuit defines a consistent process. Our main theorem is that any routed graph that has two properties, called *bi-univocality* and *weak loops*, is valid. We thus propose that in the context of indefinite causal order, 'composing quantum channels' is taken to mean combining them in accordance with a routed graph that satisfies bi-univocality and weak loops.

We emphasise that this paper does not aim to address the question of the *physical realisability* of processes (though its ideas might help tackle it in the future). Rather, we are interested in understanding the abstract, logical structure lying at the heart of valid processes with indefinite causal order. Given a valid process, we do not ask whether it could be implemented in practice, or even in principle, given the laws of physics that govern our particular universe. We ask only whether a purported quantum process is *logically consistent* (i.e. whether it implies any contradiction), what makes it so, and how it can be mathematically constructed in a way that makes this obvious.

In summary, this paper restores a notion of compositionality, along with many of its benefits, to a significant class of processes with indefinite causal order. The notion of compositionality is subtler than the one familiar from the definite order case, due to the additional rules that must be followed when composing processes. Note, however, that even in the definite order case there is a rule constraining composition: that individual compositions between processes do not amount to forming a global loop. Therefore, the introduction of restrictions on composition in this paper is not a novelty; we only propose a refinement of the commonly admitted ones.

The paper is structured as follows. We begin by introducing our framework using the example of a reconstruction of the quantum switch, in order to provide a pedagogical introduction to its main notions with a simple example. We then present our framework in full generality, which describes how to construct processes from elementary operations and their connectivity, in a way that guarantees logical consistency. Following this, we display how our framework allows us to reconstruct other processes prominent in the literature. We explain how the route structure displays the core behaviour of the processes in a compact way. We embed these examples into larger families of similar processes, and thus

highlight the conceptual intuitions for their validity. We conclude with a short discussion and outlook, in which we spell out a conjecture that all unitarily extendible processes can be built using our method.

2 Reconstructing the quantum switch

The quantum switch [8, 9] is a process with indefinite causal order whose logical consistency is relatively easy to understand. However, the obvious ways of grasping its consistency – such as recognising it as the coherent control of processes with definite causal orders – do not generalise to more exotic instances of indefinite causal order, such as the Grenoble [13] or Lugano [11, 12, 15] processes (which we will discuss later in Section 4). In this section, we reconstruct the quantum switch in a way that guarantees that it is a valid supermap[2], using a method that does generalise to more exotic unitarily extendible processes. In so doing, we sketch out the main ingredients of our framework before it is formally introduced in Section 3.

We start this section by defining supermaps, and, in particular, the supermap describing the quantum switch. We then briefly summarise some key concepts of the routed circuits framework [16, 19, 20], which we will use for the reconstruction. Finally, we perform the reconstruction in a pedagogical way.

2.1 Supermaps and the quantum switch

The quantum switch is a process in which the order of application of two transformations is coherently controlled. Mathematically, it can be represented by a *supermap*, called `SWITCH`. In the literature, a supermap is typically defined as a linear map that transforms quantum channels to quantum channels [6, 8, 9]. However, since we are only interested in unitarily extendible supermaps in this paper, we shall define a supermap as a linear map that transforms linear operators on a Hilbert space to linear operators on a Hilbert space. Given two input operators U and V of the same dimension d, `SWITCH` returns an operator of the form

$$\texttt{SWITCH}(U, V) = |0\rangle\langle 0| \otimes VU + |1\rangle\langle 1| \otimes UV\,. \tag{1}$$

More generally, U and V could be acting on their own local ancillary systems, X and Y respectively. Then `SWITCH` is defined as follows:

$$\texttt{SWITCH}(U, V) = |0\rangle\langle 0| \otimes (I_X \otimes V)(U \otimes I_Y) + |1\rangle\langle 1| \otimes (U \otimes I_Y)(I_X \otimes V) \tag{2}$$

If U and V are both unitary operators, then $\texttt{SWITCH}(U, V)$ is also a unitary operator. We call supermaps like this, that always map unitary operators to unitary operators, *superunitaries*. A little reflection reveals that any superunitary (defined for all input unitaries that can act on their own local ancillary systems) uniquely defines a supermap in the traditional sense as a map on channels via the Stinespring dilation of the channels.

An equally formal but more intuitive representation of supermaps can be provided using diagrams [5, 21]. The idea, illustrated for a monopartite supermap in Figure 1, is to represent unitary operators using boxes, and supermaps as shapes that give another box once one inserts a box into each of its 'nodes'.

[2]Note that in this paper, we will adopt the *supermap* representation of higher-order processes [6, 8, 9], as opposed to the (mathematically equivalent) *process matrix* representation [10, 15] also used in the literature. Because the relationship and equivalence between the two pictures, both at the conceptual and mathematical levels, can be a source of confusion, and in the interest of readers more accustomed to process matrices, we spell out the connection between the representations in detail in Appendix A.

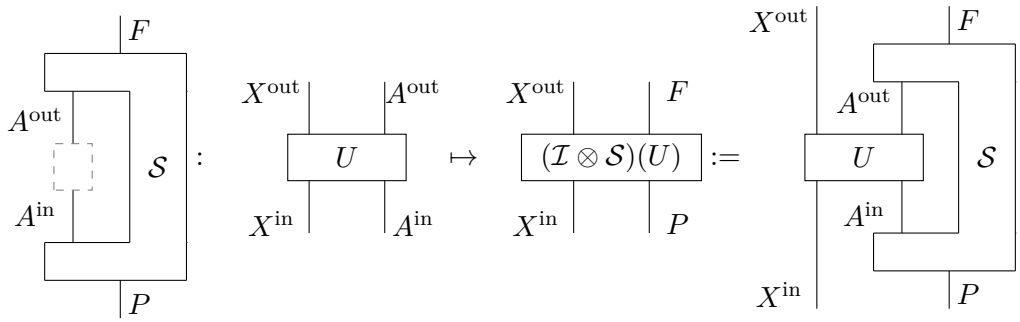


Figure 1: Diagrammatic representation of a monopartite superunitary. $\mathcal{S}$ is a linear map from unitaries of the form $U : \mathcal{H}_{X^{\text{in}}} \otimes \mathcal{H}_{A^{\text{in}}} \to \mathcal{H}_{X^{\text{out}}} \otimes \mathcal{H}_{A^{\text{out}}}$ that maps the ingoing space for any pair of ancillary systems X^{in} and X^{out} to unitaries of the form $(\mathcal{I} \otimes \mathcal{S})(U) : \mathcal{H}_{X^{\text{in}}} \otimes \mathcal{H}_P \to \mathcal{H}_{X^{\text{out}}} \otimes \mathcal{H}_F$. This definition can be extended in an obvious way for supermaps acting on an arbitrary finite number of unitary operators.

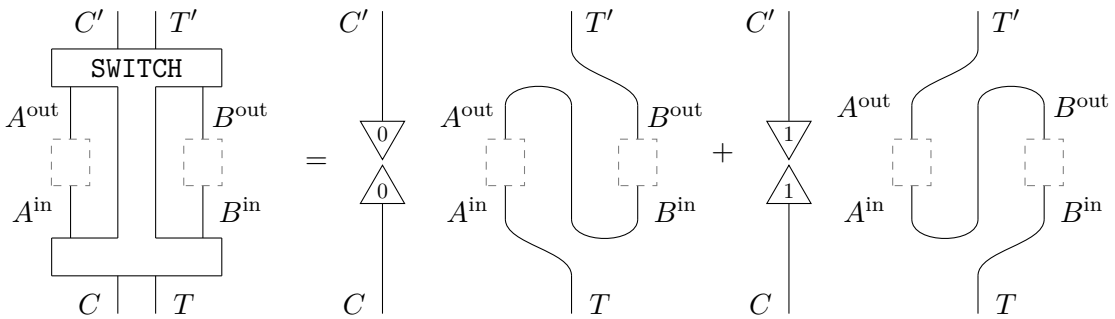


Figure 2: Circuit decomposition of the switch. The left term in the sum projects onto the $|0\rangle$ state of the control and implements Alice's transformation before Bob's. The right term has a similar interpretation. Formally, the wire bent into a 'U' shape can be interpreted as the unnormalised Bell ket $|00\rangle + |11\rangle$, and the upside-down 'U' as the corresponding bra.

In general, one can use a circuit decomposition with sums to represent a supermap. Such a decomposition is provided for the switch in Figure 2. The meaning of this circuit decomposition is that the action of the supermap is given by Figure 3, the right-hand side of which has precisely the same formal meaning as (2). However, if we do not allow for sums,[3] then it is impossible to draw $\texttt{SWITCH}(U, V)$ as a standard circuit in which both U and V appear exactly once, unless we allow feedback loops [8, 9, 22]. This means that we cannot write $\texttt{SWITCH}(U, V)$ as a standard circuit whose form makes it immediate that the switch is a consistent process.

This feature of the switch, while somewhat inconvenient, is also a hallmark of indefinite causal order. Such processes combine their input operations in a way that cannot be understood as wiring them up to form a circuit, while avoiding feedback loops. While this is precisely what makes these processes interesting, it also can leave us guessing as to what makes them consistent. If their consistency is not guaranteed by the possibility of constructing them as a standard, acyclic circuit, then what is it guaranteed by?

We will show that the consistency of the switch is guaranteed through a presentation of it as a *routed* quantum circuit [16, 19, 20]. Our method involves writing down a decorated directed graph, called the *routed graph*, which captures the basic compositional structure of

[3]This is good practice because 1) sums lead to an exponential multiplication of the number of diagrams to consider, and 2) an intuitive presentation as a sum will not be available at all in more involved cases, like the Grenoble process or the Lugano process.

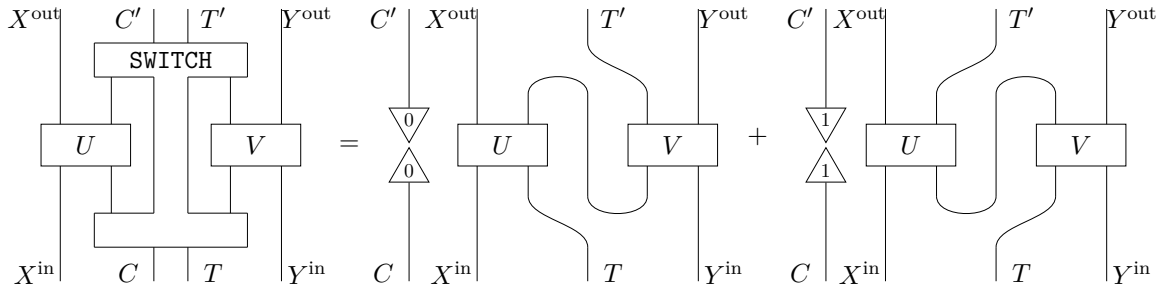


Figure 3: The action of the switch on a pair of unitary operators. The order of implementation of unitaries on the target system and ancillas is coherently controlled.

a large class of circuits. To check that all circuits that display this compositional structure are valid ones, it suffices to show that the graph satisfies two principles. Moreover, this method will generalise to other prominent unitarily extendible processes, such as Grenoble and Lugano.

All of this is in spite of the fact that the routed graph may contain feedback loops. Thus while the usual framework of quantum circuits shows how one can validly combine quantum processes in a definite causal order, our framework shows how quantum processes can be validly combined in an indefinite order.

At the heart of what makes a routed graph succeed or fail to generate consistent processes are the *sectorial constraints* it enforces at its nodes. These restrict the transformations that they accept, and, in valid graphs, they can play a vital role in outlawing those transformations that would lead to inconsistencies. But these constraints are not captured by standard quantum circuits. To handle sectorial constraints properly, we need to understand the basics of routed quantum circuits, to which we now turn.

2.2 Routed quantum circuits

We start our summary of the framework of routed quantum circuits by considering a simple physical scenario that is conveniently represented with routed circuits. Suppose a photon with an internal degree of freedom is sent to a superposition of two paths, either going to the left or to the right depending on the logical value of a control qubit C.

This can be represented with the following transformation $W : \mathcal{H}_C \otimes \mathcal{H}_T \to \mathcal{H}_L \otimes \mathcal{H}_R$, where T is the internal degree of freedom of the photon.

$$\begin{aligned} W(|0\rangle \otimes |\psi\rangle) &= |\psi\rangle \otimes |\text{vac}\rangle \\ W(|1\rangle \otimes |\psi\rangle) &= |\text{vac}\rangle \otimes |\psi\rangle \end{aligned} \tag{3}$$

Each of the output spaces $\mathcal{H}_L$ and $\mathcal{H}_R$ is the direct sum of a vacuum sector spanned by $|\text{vac}\rangle$ and a single-particle sector with the dimension of the target space $\mathcal{H}_T$ [23]: $\mathcal{H}_L = \mathcal{H}_L^{\text{vac}} \oplus \mathcal{H}_L^{\text{par}}$ and $\mathcal{H}_R = \mathcal{H}_R^{\text{vac}} \oplus \mathcal{H}_R^{\text{par}}$.

The physical evolution here is clearly reversible; yet, W is not a unitary operator, because its output space, $\mathcal{H}_L \otimes \mathcal{H}_R = \bigoplus_{i,j\in\{\text{vac},\text{par}\}} \mathcal{H}_L^i \otimes \mathcal{H}_R^j$, is 'too large'. To fix this, we could consider an extension of W to a unitary acting on a larger input space. However, this would require us to consider additional, possibly physically irrelevant degrees of freedom just to represent our original scenario, even though this original scenario was already reversible.

Another solution is to restrict the definition of W to the subspace of its output space that can actually be populated – this is the one-particle subspace $\mathcal{H}^{\text{prac}} := (\mathcal{H}_L^{\text{par}} \otimes \mathcal{H}_R^{\text{vac}}) \oplus (\mathcal{H}_L^{\text{vac}} \otimes \mathcal{H}_R^{\text{par}})$. The resulting operator $\tilde{W} : \mathcal{H}_C \otimes \mathcal{H}_T \to \mathcal{H}^{\text{prac}}$ is indeed unitary, but it

cannot be represented as a standard circuit with one output for L and another for R. The reason is that the formal meaning of putting two wires next to each other in a standard circuit is taking the tensor product of the associated Hilbert spaces, while our $\mathcal{H}^{\text{prac}}$ is composed from $\mathcal{H}_L$ and $\mathcal{H}_R$ via direct sums as well as tensor products.

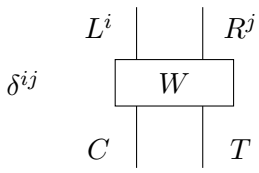
In order to achieve a better circuit representation, we can label the output subspaces in the following way.

$$\begin{aligned} \mathcal{H}_{L^{\text{par}}} &:= \mathcal{H}_L^0 \\ \mathcal{H}_{L^{\text{vac}}} &:= \mathcal{H}_L^1 \\ \mathcal{H}_{R^{\text{par}}} &:= \mathcal{H}_R^1 \\ \mathcal{H}_{R^{\text{vac}}} &:= \mathcal{H}_R^0 \end{aligned} \tag{4}$$

Then the physically relevant output space $\mathcal{H}^{\text{prac}}$ is the direct sum of the subspaces $\mathcal{H}_L^i \otimes \mathcal{H}_R^j$ for which the indices match. This is equivalent to saying that W *follows the route* given by the Kroenecker delta δ^{ij}. This just means that it respects the equation

$$W = \sum_{ij} \delta^{ij} \cdot (\pi_L^i \otimes \pi_R^j) \circ W \,, \tag{5}$$

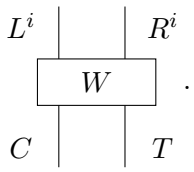
where the π_L^i and π_R^j are projectors onto $\mathcal{H}_L^i$ and $\mathcal{H}_R^j$ respectively. Now, we can formally represent our transformation as a unitary by adding indices to the outputs that represent the sectorisation of the Hilbert spaces, and decorating W with the route matrix δ^{ij}. This gives the following diagram.



(6)

The interpretation of this diagram is that we have a transformation W which follows the route δ. The route matrix tells us that W only maps states to its so-called *practical output space*, which is now defined via the route as $\mathcal{H}_{\text{prac}} := \bigoplus_{ij} \delta^{ij}\, \mathcal{H}_L^i \otimes \mathcal{H}_R^j$. W is unitary with respect to this output subspace, so the routed map (δ, W) is called a *routed unitary* transformation, even though W is not strictly speaking a unitary operator.

We can simplify the diagram (6) by introducing a shorthand called *index-matching*. Since the effect of δ^{ij} is just to 'match up' the value of the output indices in the practical output space, we can avoid writing the matrix explicitly and instead just match the output indices directly:



(7)

We now explain the notion of a routed linear map in full generality. Given a transformation $U : \mathcal{H}_A \to \mathcal{H}_B$, we can construct a routed transformation (λ, U) by first sectorising U's input and output Hilbert spaces into a set of orthogonal subspaces:

$$\begin{aligned} \mathcal{H}_A &= \bigoplus_i \mathcal{H}_A^i \\ \mathcal{H}_B &= \bigoplus_j \mathcal{H}_B^j \end{aligned} \tag{8}$$

We then specify a Boolean matrix λ, which is there to tell us which $\mathcal{H}_A^i$ may be mapped to which $\mathcal{H}_B^j$. We call this sort of restriction on the form of U a *sectorial constraint*. Specifically, U *follows* λ if it satisfies

$$U = \sum \lambda_i^j \cdot \pi_B^j \circ U \circ \mu_A^i \tag{9}$$

where the π_B^j's and μ_A^i's project onto sectors $\mathcal{H}_B^j$ and $\mathcal{H}_A^i$ respectively. For (λ, U) to be a valid routed transformation, we need U to follow λ.

The routes can be composed in parallel and in sequence by the Cartesian product and matrix multiplication, respectively. Routed maps can be composed by composing their elements pairwise: the composition of the linear maps will necessarily follow the composition of the routes. This enables us to build up large routed circuits using elementary routed maps as building blocks.

The route λ defines U's practical input and output spaces $\bigoplus_i (\sum_j \lambda_i^j) \mathcal{H}_A^i$ and $\bigoplus_j (\sum_i \lambda_i^j) \mathcal{H}_B^j$ respectively. U is a routed unitary if the transformation is unitary when we restrict its definition to these spaces.

The last thing to introduce is the notion of a *routed supermap* [19]. In a standard superunitary, any unitary operator that maps the ingoing space $\mathcal{H}_{A^{\text{in}}}$ to the outgoing space $\mathcal{H}_{A^{\text{out}}}$ of one of the nodes is considered a valid input to that node. For a routed supermap, a node might be equipped with some route λ given a sectorisation of its input and output spaces. Then, the only valid unitary transformations for that node are those that follow λ. Formally, the valid inputs are those that respect $U = \sum_{ij} \lambda_i^j \cdot \mu_B^j \circ U \circ \pi_A^i$, or $U = \sum_{ij} \lambda_i^j \cdot (\mu_B^j \otimes I_X) \circ U \circ (\pi_A^i \otimes I_X)$ if the unitary also acts on some ancillary system X.

2.3 Extracting the relevant structure: routed circuit decomposition, skeletal supermap, routed graph

Armed with an understanding of routed maps, we can now give the promised routed circuit decomposition of the switch. Luckily, all we really need is a routed unitary of the form (δ, W), represented in (7).

Our decomposition of the switch is presented in Figure 4. The basic intuition behind the diagram lies in the following interpretation. When we prepare the control qubit in $|0\rangle$, the target system may only enter the sectors of the wires inside the diagram corresponding to $i = 0$. Recalling that $\mathcal{H}_L^0$ was a d-dimensional sector, and that $\mathcal{H}_R^0$ was a trivial, one-dimensional 'dummy sector', this means that the particle will exit via the left output port of every W it enters. Meanwhile, the right output will receive a one-dimensional dummy system, analogous to the vacuum in the interferometric example above (although this is merely a formal analogy – we are not committing to any particular physical interpretation of this dummy system). This means that the particle will go through Alice's node first, then Bob's, then out to the future. The opposite is true when we prepare the control in $|1\rangle$.

We now want to nail down how the route structure in Figure 4 can be leveraged to certify that the supermap is a valid one. To do this, we first need to to consider a further pruned version of the circuit, in which only the essential information appears. This is given by what we call a *skeletal supermap*: a supermap that includes nothing other than wires, without any boxes representing non-identity unitary transformations. The idea is that we can obtain the original supermap from the skeletal supermap by 'fleshing it out', i.e. inserting some unitary transformations into the nodes. If we can show that this skeletal

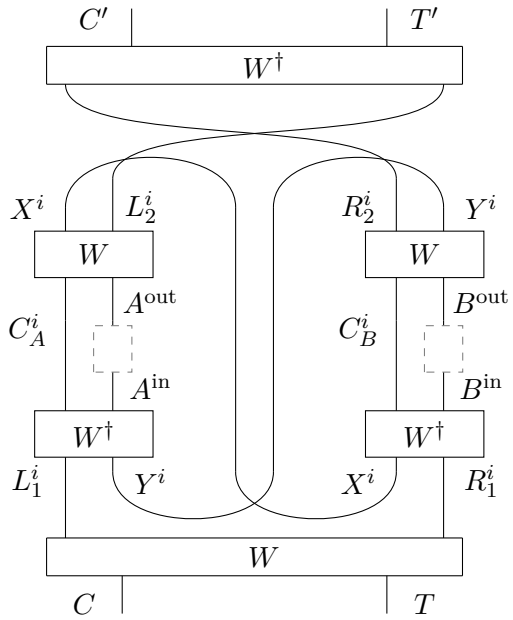
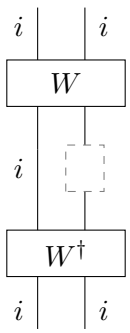


Figure 4: Routed circuit decomposition of the switch, using index matching. W is the routed unitary defined in (3). The wires bent into 'cup' or 'cap' shapes represent the (unnormalised) perfectly correlated entangled states [5]. Overall, the '$i = 0$' sectors correspond to the branch where Alice's intervention is implemented before Bob's, while '$i = 1$' corresponds to the branch where Bob's intervention is implemented before Alice's. Thus the cycle is constructed from two acyclic components corresponding to definite orders of implementation.

supermap is a valid superunitary, then it follows immediately that our original supermap is a valid superunitary.

A suitable skeletal supermap for the switch is represented in Figure 5. Inserting W and $W^\dagger$ into the bottom and top nodes respectively, and inserting the superunitary



(10)

into each of the middle nodes yields `SWITCH`.

We can represent the skeletal supermap using an even simpler graph. All we need to consider is the connectivity between the nodes, the routes and indices, and the specific index values that represent one-dimensional sectors. A representation of all this information, and nothing more, is provided by the *routed graph*. This consists of

- a vertex for each node in the skeletal supermap, decorated with its route;
- arrows representing the wires connecting the nodes in the skeletal supermap;

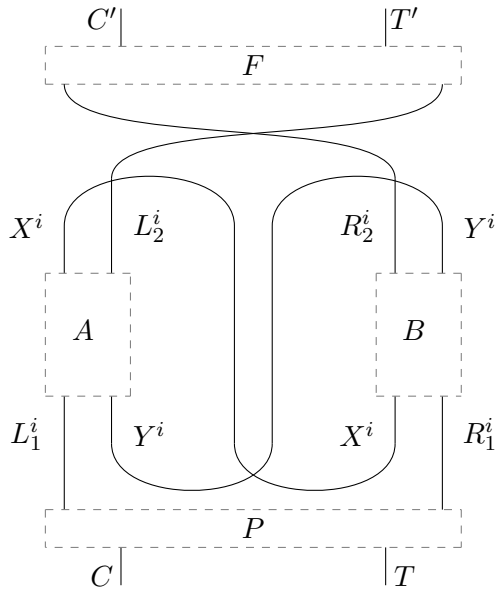


Figure 5: Skeletal supermap for the switch. The nodes suffer sectorial constraints represented by the index-matching of the input and output wires, making this a *routed* supermap. The routed circuit for the switch in Figure 4 is obtained by inserting unitary transformations into the nodes P and F, and the monopartite superunitary (10) into A and B.

- next to each arrow, the index of the corresponding wire;
- next to each arrow, the specific values of its index that corresponds to a one-dimensional sector.

When a node has a 'delta route' – that is, a route that is equal to 1 if and only if all the indices take the same value – we can adopt the shorthand index-matching representation where we decorate each of its ingoing and outgoing arrows with the same index.

The routed graph for the switch's skeletal counterpart is given in Figure 6, with and without the index-matching shorthand. Remarkably, this elementary object contains all the information we need to confirm that the switch is a valid superunitary, or in other words, that it is consistent.

2.4 Checking for validity

In our framework, one can just consider the routed graph depicting the connectivity of the supermap, and infer from it that the supermap is valid. This amounts to checking that the routed graph conforms to a couple of principles. Here we shall present these principles and the way to check them in a pedagogical manner, taking advantage of the relative simplicity of the switch's case.

To motivate these principles, a good place to start is with the intuition that in a self-consistent protocol, information should not genuinely be able to flow in a circle. This is because, if it did, then at any point on the circle we could control the outgoing information on incoming information that is inconsistent with it. This happens in the grandfather paradox, where Alice's grandfather is killed if Alice exists, even though Alice's existence is incompatible with his murder.

Yet from the present routed graph it seems as if information does flow in a circle between A and B. What we need to do is use the information in the graph to obtain a

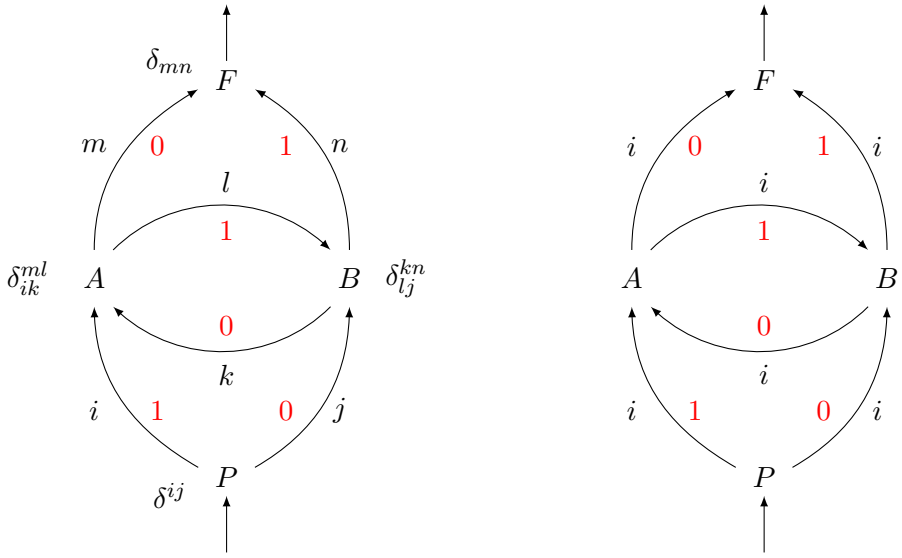


Figure 6: Routed graph for the switch, with and without index-matching. The vertices represent nodes from the skeletal supermap. On the left, each wire is equipped with its own index, and the numbers in red denote the values of the index that correspond to a one-dimensional sector. Each node is decorated with a δ matrix representing the route, which is equal to 1 if and only if all of its arguments are equal. Lower indices refer to input wires, upper indices refer to output wires. Since all the routes are 'delta' routes, we can use the convenient shorthand of index-matching to produce a simpler diagram with the same meaning, as on the right.

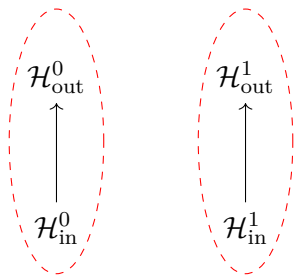


Figure 7: The route for the A node of the skeletal supermap.

more fine-grained perspective from which the cycle disappears (or is at least shown to be harmless).

We start by fine-graining each node into a number of *branches*. If the route of a node dictates that there are exactly n disjoint subspaces of the input space that must be mapped one-to-one to n disjoint regions of the output space, we say that there are n branches.

To make this clear, we can represent the route matrices as diagrams with arrows from the input sectors $\mathcal{H}^i_{\text{in}}$ to the output sectors $\mathcal{H}^j_{\text{out}}$ being present when the corresponding route matrix element λ^j_i is equal to 1. For the node A, we have a route of the form δ^j_i, which is represented in Figure 7. In this sort of diagram, each disconnected 'island', circled in red, corresponds to a distinct branch. Thus A has two distinct branches, which we label A^i in correspondence with the value of i. On the other hand, although the node P's outgoing space has two sectors, P only has one branch, since its graph is fully connected, as represented in Figure 8.

Intuitively, branches correspond to *alternatives* in a node: for example, in node A,

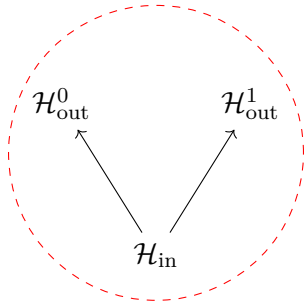


Figure 8: The route for the P node of the skeletal supermap.

either the branch A^0 or the branch A^1 will happen.[4] It is these branches, rather than the original nodes, that we will study. In particular, we will check whether the branches themselves form informational loops; this will be the subject of our second principle.

2.4.1 Bifurcation choices and bi-univocality

Before proceeding to investigate informational loops, we first need to check another, more fundamental, property. As we remarked, branches correspond to alternatives. Thus, roughly speaking, we should make sure that the route structure is detailed enough to specify that exactly one branch happens at each node. This is captured by the notion of a *bifurcation choice*.

Let us introduce it with an example, resorting to agents for intuition. In the route for the P node in Figure 8, the ingoing space may be mapped to two different sectors of the outgoing space, meaning that an agent can 'choose' to send information to just one of these sectors. More generally, an agent at a node can make a 'bifurcation choice' for each branch of that node (for the branches that contain only one output value, the choice is trivial).

In the routed graph of Figure 6, only P features a bifurcation choice. Furthermore, this bifurcation choice amounts to picking the value of the index i through the graph; thus, if the agent at P picks, say, $i = 0$, this leads (through the other routes) to that value getting instantiated through the graph, and therefore to the branches A^0 and B^0 'happening'. The symmetric situation happens for the $i = 1$ choice. In other words, each possible bifurcation choice determines exactly one branch to happen at every node.

It is this behaviour, rather elementary in the case of the switch, that we want to ask for in general. This leads to a principle that we will call *univocality*:[5] any tuple of choices made at every branch leads to exactly one branch happening at every node. In other words, once the agents at the nodes of our skeletal supermap make all their bifurcation choices, there is a determinate fact, for each branch, about whether the quantum state will pass through it. More formally, this will be defined as the fact that the routed graph defines a function (as opposed to a relation) from bifurcation choices to 'branch statuses', where branch statuses are bits representing whether a given branch has happened or not.

[4]Of course, because we are in quantum theory, both could happen in a superposition. But a remarkable feature of our framework is that, in order to check the validity of the routed graph, we do not have to consider superpositions: it is sufficient to reason as if the branches were mutually exclusive. Therefore this is what we will do through this section.

[5]This is a shameless gallicism. 'Univocal' means 'speaking with one voice', i.e., yielding exactly one output. For instance, functions are univocal, while relations (represented here by Boolean matrices) are generically equivocal.

(Section 3 describes how this function is defined.) This can be seen as forbidding situations where bifurcation choices would either underdetermine branch statuses (i.e. lead to several possible branch assignments) or overdetermine them (i.e. lead to no possible assignment at all).[6]

For the switch, this is satisfied because the bifurcation choice at the P node of the skeletal supermap determines which branches of A and B we end up in. This corresponds to the fact that in `SWITCH`, the logical state of the control qubit fixes the causal order (recalling the fact that the causal order is what defined the different branches of A and B).

We also require that the 'time-reversed' routed graph, obtained by reversing the direction of the arrows on the original routed graph, satisfies univocality as well. This is satisfied by the switch, corresponding to the fact that the information about which causal order took place ends up recorded in the control qubit at the end of the protocol. If both the routed graph and its time-reversed version satisfy univocality, we say that the routed graph satisfies **bi-univocality**. Thus the entire bi-univocality condition is satisfied by `SWITCH`. We summarise the condition as follows:

> *Bi-univocality:* The routed graph and the time-reversed routed graph define functions from bifurcation choices to branch statuses.

2.4.2 The branch graph and weak loops

We now turn to our second principle, which deals with whether influences between branches flow in a circle. To check this, we construct a directed 'branch' graph representing the flow of information between different branches in the routed graph, depicted in Figure 9. Causal/informational loops among the branches will be understood as loops in this graph.

The branch graph contains solid, dashed green, and dashed red arrows. The solid arrows represent the flow of quantum information along 'paths' in the routed graph permitted by the routes, while the dashed arrows represent the flow of information via choices of which path to follow, when multiple paths are permitted by the routes. We explain each of these in turn at an intuitive level; the general formal procedure for constructing the branch graph from a routed graph is described in Section 3.

To understand the solid arrows, note that there are two possible joint value assignments to all of the indices in the routed graph: either $i = 0$ everywhere, or $i = 1$. For the $i = 0$ assignment, the arrows $P \to B$ and $B \to A$ in Figure 6 correspond to one-dimensional sectors, as indicated by the red zeroes. What this shows is that no quantum information flows from P to the branch B^0 or from B^0 to A^0. For this reason, there are no solid arrows $P \to B^0$ or $B^0 \to A^0$ in the branch graph. On the other hand, quantum information does flow from P into the branch A^0, then into B^0, and then finally into F. Thus we have the path $P \to A^0 \to B^0 \to F$ of solid arrows in the branch graph. By following precisely analogous reasoning for the $i = 1$ assignment, we arrive at the solid arrows in Figure 9. Evidently, the solid arrows in the branch graph do not form a loop[7].

[6] On the relationship between underdetermination and overdetermination in cyclic processes, see Ref. [24].

[7] We note that this corresponds to an observation from Ref. [25], that, although the switch has a cyclic causal structure, it can still be written as a direct sum of (pure) processes with a definite causal order. We want to stress however that such an observation is in general *not* sufficient to ensure the consistency of the process, as it overlooks the need to 1) check bi-univocality, and 2) also represent dashed arrows in the branch graph.

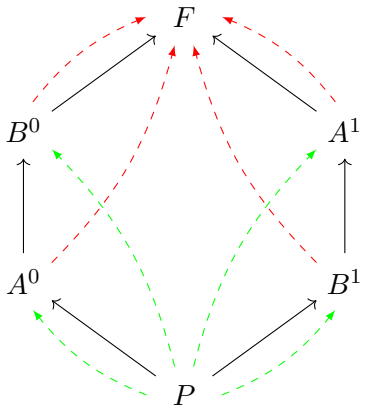


Figure 9: The branch graph for the routed graph in Figure 6. Each vertex represents the branch of some node in the routed graph. The branches of A and B are labelled with superscripts corresponding to the relevant value of i in the index-matching routed graph. The other nodes have only one branch, and we denote this branch with the same letter we used for the original nodes. The solid arrows are attributed by considering the connection between the branches encoded in the routed graph; the dashed green and red arrows represent relations of functional dependence in the functions from bifurcation choices to branch statuses required by bi-univocality. The graph contains no cycles of any kind, so it trivially satisfies the weak loops condition.

To rule out informational loops, it is necessary that the solid arrows do not form a loop, but it is not sufficient. What the lack of this kind of loop shows is that the quantum information confined within particular branches by the routes does not flow in a circle. But there is another type of information flowing in the routed circuit: the information that determines *which branch happens*. This information is represented by the dashed arrows in Figure 9. It is possibilistic in nature, and can therefore be captured entirely using routes, based on the theory of finite relations (i.e. Boolean matrices).

Fortunately, if univocality is satisfied, then we already know that the routed graph defines a function from bifurcation choices to the statuses of the branches (i.e. the binary variables encoding whether each branch happened). We can thus define the green dashed arrows as representing the functional dependencies within this function. Namely, there is a green dashed arrow from X^α to Y^β just in case the branch status at Y^β depends on the bifurcation choice at X^α. For example, there is a green dashed arrow from P to A^0 because one can choose whether or not A^0 happens by choosing which logical state to prepare the control qubit in at P. If a similar influence relation holds from Y^β to X^α in the time-reversed version of the protocol, then we draw a red dashed arrow from X^α to Y^β. Doing this for all the branches gives us the dashed arrows in Figure 9.

The full branch graph gives a complete account of the flow of information in the skeletal supermap of the switch. It represents both the quantum information that flows within the branches with the solid arrows, and the 'which-branch' information that is affected by bifurcation choices. This second sort of information can be thought of classically, since it corresponds to preferred sectorisations of the state spaces. We also call it possibilistic, since it is purely about the binary question of whether a branch does or does not happen given certain bifurcation choices, and can accordingly be represented by the routes using the theory of relations, represented by Boolean matrices.

From this fine-grained perspective, it is clear that no information actually flows in a loop in `SWITCH`, since the branch graph of Figure 9 satisfies

No loops: There are no directed loops in the branch graph.

According to the upcoming Theorem 3.1 – the main theorem of this paper –, the satisfaction of bi-univocality and no loops is enough to demonstrate the validity of the skeletal supermap, and hence of the switch itself. Thus the logical consistency of the switch is a consequence of the satisfaction of these principles.

Remarkably though, a principle logically weaker than no loops is enough to ensure the validity of the supermap. We did not need to show that the routed graph contains no loops at all, but only loops of a weak, and harmless, type. Specifically, we needed to show that the graph satisfies the following principle, which we call **weak loops**.

> *Weak loops:* Any given loop in the branch graph is entirely made up of dashed arrows of a single colour.

Our main theorem states that bi-univocality together with weak loops implies that a skeletal supermap is valid, and hence that any associated protocol is self-consistent. While all protocols we have studied that do not violate causal inequalities satisfy no loops, Section 4.3 will show that the Lugano process has green loops (see Figure 29). This will lead us to conjecture that *the presence of monochromatic loops is precisely what enables the violation of causal inequalities.*

Some readers might be familiar with another routed circuit for the quantum switch, namely the one in Figure 7 of [25]. Let us briefly explain why we have used a different circuit here. Suppose we started with Figure 7 of [25] and followed the same procedure that we have followed in this section, removing the unitaries in the circuit to obtain a skeletal graph, and then using the skeletal supermap to write down a routed graph. That routed graph is *not* a valid one.

The fastest way to see this is to note that in Figure 7 of [25], there is a loop made up entirely of wires that have no index, and hence a trivial sectorization. This will result a loop made up entirely of black wires in the routed graph, and thus a violation of the weak loops condition. Of course, the routed circuit in [25] describes a perfectly valid process – namely the switch – but the process is only valid because of the *specific* unitaries used in the circuit, and not as an inevitable consequence of the way that those unitaries are combined.

2.5 Why do we need bi-univocality?

Naively, one might imagine that the lack of a causal/informational loop among the branches is enough to guarantee that a protocol is consistent. In this subsection, we explain why this intuition fails.

Let us first explain why univocality is important. To this end, consider the supermap in Figure 10. A single wire is bent round in a loop, and serves both as input and as output to a node. The wire represents a qubit partitioned into sectors spanned by $|0\rangle$ and $|1\rangle$ respectively. We impose a delta-route on the node so that the transformations we insert must map each sector to itself.

The node has two branches, each corresponding to a one-dimensional sector. Hence, no information flows between the branches. Clearly, then, there can be no question of an informational loop within a branch. Nevertheless, the supermap is invalid, and fails to represent any logically consistent process. This is a consequence of the fact that *nothing can determine the value of the index i.* For if it did, then univocality would be respected, and our main theorem would imply the validity of the supermap.

The key problem here is one of underdetermination. The fact that i is not fixed means that there is no point in the circuit where an agent could make a bifurcation choice determining which branch happens. So, even though there is no *over*determination *within*

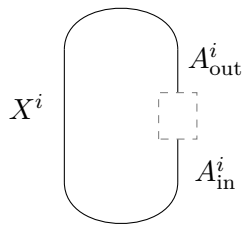


Figure 10: A routed skeletal supermap that leads to a paradox, where a qubit is sent back to itself. Formally, the wires bent into 'cup' and 'cap' shapes can be thought of as the unnormalised perfectly correlated entangled ket and bra respectively. The qubit is partitioned into sectors spanned by the logical states $|0\rangle$ and $|1\rangle$. The index-matching means that the agent at the node must map the logical states $|i\rangle$ to themselves, up to dephasing.

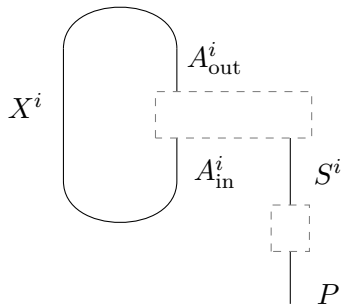


Figure 11: A non-valid routed skeletal supermap, leading to a paradox. This supermap is similar to that of Figure 10, but now includes an S qubit system, partitioned into two one-dimensional sectors, and an unsectorised P qubit system.

a branch (resulting in a grandfather paradox), there is still *under*determination *of* the branch. To see how this leads to concrete problems, one needs only to notice that inserting a CNOT gate into the supermap (where the NOT part acts on an ancillary system) leads to a trace-decreasing output transformation, and that this is a direct consequence of the underdetermination of i.[8]

Let us now illustrate why not only univocality, but *bi*-univocality is necessary for the consistency of a protocol. To this end, consider the supermap in Figure 11. In contrast with the previous example, this supermap includes an additional node in which 'i is created'; the corresponding routed graph therefore satisfies the univocality principle. However, this supermap is not a valid one: plugging an identity in the $P \to S^i$ node and the routed unitary $|ii\rangle \mapsto |i\rangle$ in the other node yields the $\langle 0| + \langle 1|$ effect, which is of course not a unitary. This failure can be ascribed to the fact that the routed graph corresponding to this supermap does not satisfy *bi*-univocality: its adjoint does not satisfy univocality. In broad terms, not only should the indices 'come from somewhere', they should also 'go somewhere'.[9]

Before moving on to lay out our framework in the next section, let us first summarize this one, in which we have shown how to represent the switch as a routed circuit and how

[8]The reader might also notice that inserting a Z Pauli matrix into the node results in a sort of grandfather paradox associated with the Fourier basis. But the foregoing considerations show that even in the classical case, where there is no Fourier basis, we still need bi-univocality.

[9]One might naively believe that only requiring univocality would lead to super*isometries* that would not necessarily be superunitaries, but the above example also shows this is not the case, since the supermap in Figure 11 is not even a superisometry.

to certify the validity of this circuit despite its feedback loops.

We wrote the switch as a routed circuit (Figure 4). We captured this circuit's basic structure by trimming it down to a 'skeletal' routed supermap (Figure 5), from which the switch can be constructed, by inserting unitary transformations into the nodes. We represented the structure of the skeletal supermap as an equivalent routed graph (Figure 6). We then showed that this routed graph satisfies two conditions, bi-univocality and weak loops, which, by our main theorem, imply that the skeletal supermap is valid (i.e. takes unitaries to unitaries), which in turn implies that any routed circuit with its connectivity is valid as well.

Bi-univocality requires that choices of bifurcation in the routed graph lead to a definite fact about the branch that happens at each node. It also requires a similar statement to hold about the time-reversed version of the routed graph, obtained by reversing the direction of the arrows.

If bi-univocality holds, then we can ask whether the routed graph satisfies the weak loops condition. To evaluate this condition, we form a branch graph, in which solid arrows represent the ability of quantum information to flow between the different branches. Green dashed arrows indicate that bifurcation choices at one branch can influence whether another branch happens in the routed graph. Red dashed arrows represent the same thing for a time-reversed version of the routed graph. The weak loops condition states that any given loop in the branch graph must be formed entirely of dashed arrows of the same colour. The switch satisfies this trivially since its branch graph contains no loops all.

Our constructions do not only provide a technical way to certify the consistency of a process, but also make its inner structure evident. Indeed, the routed graph gives an intuition of the crucial structural features of the switch. Furthermore, its branch graph displays the order in which its branches happen, and tells us which branches control which other branches happen. This will be particularly valuable when we perform the same reconstruction for more elaborate processes in Section 4.

3 The framework

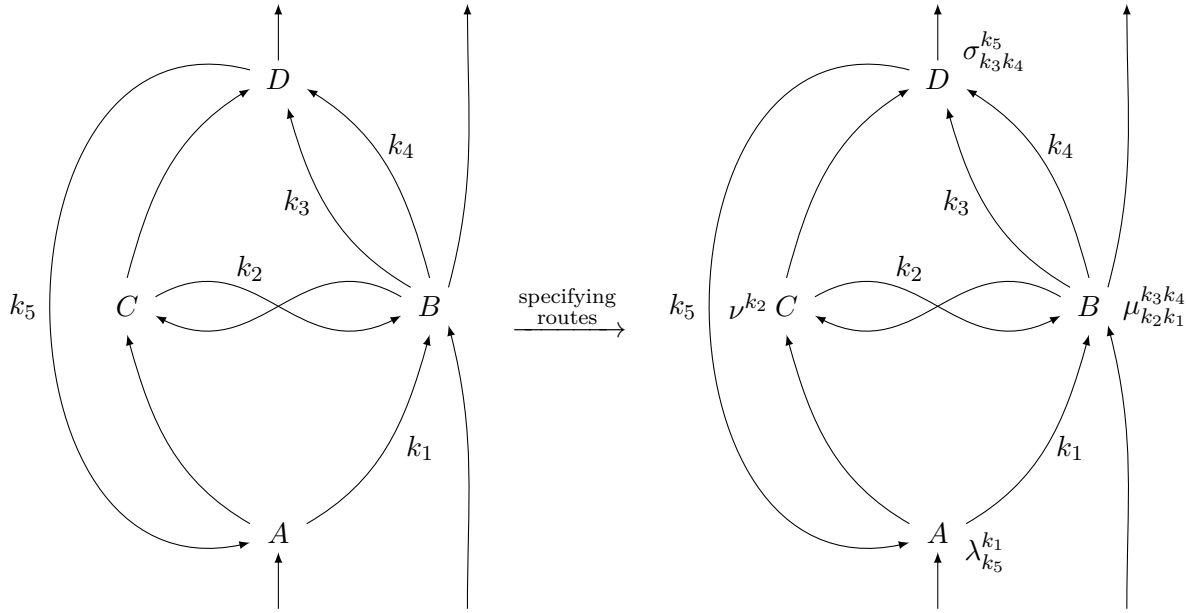
In this section, we present our framework in detail and state our main theorem, which says that any routed graph satisfying bi-univocality and weak loops defines a valid superunitary. To keep things readable, we will give definitions at a semi-formal level; a fully formal account is given in Appendix B.

The most basic notion we need is that of a routed graph: this is a directed multi[10]-graph with decorated nodes and arrows. The nodes are decorated with routes, and the arrows are decorated with indices that are in turn equipped with a 'dimension' for each index value. A routed graph with its routes still unspecified will be called an indexed graph. Examples are given in Figure 12.

Definition 3.1 (Indexed and routed graphs)**.** *An* <u>*indexed graph*</u> Γ *is a directed multigraph in which each arrow is attributed a non-empty set of index values. Each of these values is furthermore attributed a non-zero natural number, called its dimension*[11]*.*

[10] A multigraph is a graph in which there can be several arrows between two given nodes. In the interest of generality, we will allow them, even though for the purposes of the certification of supermaps' validity, any multigraph could just be turned into an equivalent graph by merging wires.

[11] This will be the dimension of the corresponding sector in the interpretation of the graph as a supermap. Note that for our theorem, all we need to know is which sectors are one-dimensional.



(a) An indexed graph. (b) A routed graph.

Figure 12: Examples of an indexed graph and of a routed graph; the latter is obtained from the former by specifying a branched route at every node. The arrows not bearing indices have a trivial (i.e. a singleton) set of index values.

A <u>*routed graph*</u> $(\Gamma, (\lambda_N)_{N \in \mathtt{Nodes}_\Gamma})$ *is an indexed graph for which a relation (or 'route') has been specified at every node. The route* λ_N *at node* N *goes from the Cartesian product of the sets of indices of the arrows going into* N*, to that of the sets of indices of the arrows going out of* N*.*

We also allow these graphs to feature arrows 'coming from nowhere' (resp. 'going nowhere'): these will be interpreted as global inputs (resp. global outputs) of the supermap. We ask for these not to be indexed, that is, to have trivial (i.e. singleton) sets of index values.[12]

We will in fact not need all types of relations: we will restrict ourselves to considering *branched* ones.

Definition 3.2 (Branched routes)**.** *A route* λ_N *is* <u>*branched*</u> *if any two of its input values are either connected to the exact same output values, or have no output values in common.*

An example is given in Figure 13. As seen in this figure, a branched relation λ_N defines compatible (partial) partitions of its input and output sets, which we call λ_N's *branches* (or, in a slight abuse of notation, N's branches, which will be called the N^α with α varying), with each input value of a branch being connected to all output values of this branch and vice versa.

There can also be input (resp. output) values that are not connected to anything by λ_N; these will be said to be outside its practical inputs (resp. outputs), and are considered to be part of no branch at all. These values correspond to sectors which are just there for

[12]This requirement is there only to make the statement of univocality simpler, as otherwise one would have to distinguish several cases. Any routed graph with indexed input and output arrows can be turned into one without, by adjoining to it a global input node and a global output node.

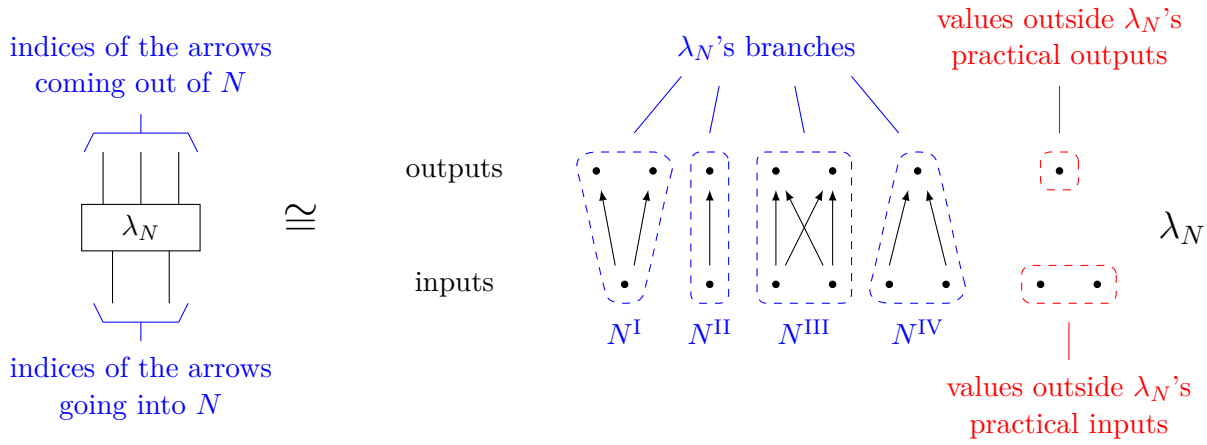


Figure 13: 'Looking inside' a branched route λ_N (which is to be used for the node N of an indexed graph). On the left, we see λ_N as a box: each of its input (resp. output) wires is the set of index values of one of the arrows going into N (resp. coming out of it). On the right, we see its 'unfolded' structure, specifying how λ_N connects input values to output values; each of the input black dots corresponds to a possible value (more precisely, a possible tuple of values) of the input indices, and similarly for the outputs.

formal purposes and will never be used in practice – part of the role of bi-univocality will be to ensure that this does not lead to any inconsistencies.

A skeletal routed supermap can be naturally defined from a routed graph.

Definition 3.3 (Skeletal supermap associated to a routed graph)**.** *Given a routed graph* $(\Gamma, (\lambda_N)_N)$*, its associated skeletal (routed) supermap is obtained by interpreting each wire as a sectorised Hilbert space, whose sectors are labelled by the set of index values of this wire, with each sector having the dimension that was assigned to its corresponding index value; and interpreting each node as a slot for a linear map, going from the tensor product of the Hilbert spaces associated to its incoming arrows, to that of the Hilbert spaces associated to its outgoing arrows, and following the route associated to that node. The supermap acts on linear maps by connecting them along the graph of* Γ[13]*.*

Our goal is to define structural requirements on routed graphs ensuring that their associated supermap is a (routed) *superunitary*; i.e., that it yields a unitary map when arbitrary unitary maps, following the routes, are plugged at each of its nodes. Note that a map being unitary, in this context, means that it is unitary when restricted to act only on its practical input space, consisting of the input sectors whose indices are practical inputs of the route, and to map to its similarly defined practical output space.

Our first principle will be univocality. The idea is that some branches feature *bifurcations*, i.e. include several output values (e.g. branches N^{I} and N^{III} in Figure 13). 'Bifurcation choices', in a branch at a node – i.e. choosing a single output value for this branch, and erasing the arrows to the other output values – will in general lead to some branches at other nodes 'not happening' – i.e. to none of their input values being instantiated. Univocality tells us that *any tuple of bifurcation choices* throughout the graph should lead to *one and exactly one* branch happening at every node. To make this requirement formal, we will 'augment' our relations, i.e. supplement them with ancillary wires: ancillary input wires with which bifurcation choices in each branch can be specified; and

[13]Note that this procedure has an unambiguous meaning, despite the cycles in Γ, due to the fact that finite-dimensional complex linear maps form a traced monoidal category [26].

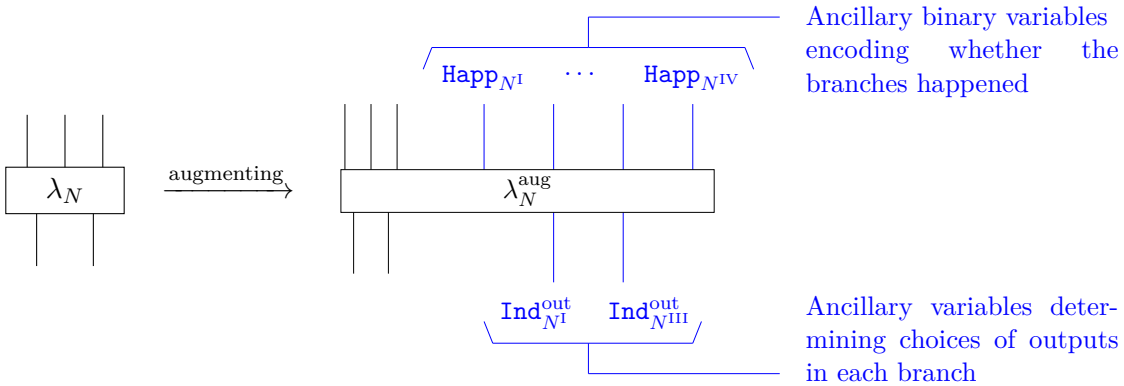


Figure 14: The 'augmented' version of the branched route λ_N described in Figure 13. The ancillary input wires for branches N^{II} and N^{IV} are not written, as they are trivial: each of these branches has only one output value.

ancillary output wires which record, in a binary variable, whether each branch happened or not.

Definition 3.4 (Augmenting)**.** *We take a branched route λ_N. For each of its branches N^α we denote the set of output values of this branch as $\mathtt{Ind}^{\mathrm{out}}_{N^\alpha}$, and define a binary set $\mathtt{Happens}_{N^\alpha} \cong \{0,1\}$.*

The augmented version λ_N^{aug} of λ_N is the partial function going from λ_N's input values and from the $\mathtt{Ind}^{\mathrm{out}}_{N^\alpha}$'s, to λ_N's output values and the $\mathtt{Happens}_{N^\alpha}$'s, defined in the following way:

- *if its argument from λ_N's input values is among the input values of a branch N^α, then it returns its $\mathtt{Ind}^{\mathrm{out}}_{N^\alpha}$'s argument, value 1 in $\mathtt{Happens}_{N^\alpha}$, and value 0 in $\mathtt{Happens}_{N^{\alpha'}}$ for $\alpha' \neq \alpha$;*
- *if its argument from λ_N's input values is not among the input values of any branch – i.e. if it is outside of λ_N's practical input values –, then the output is undefined.*

To illustrate this definition, let us display what it gives in the case of the routed graph of the switch, depicted in Figure 6. The augmented version of P's route has one extra binary input (encoding the bifurcation choice that can be made at this node), and no extra output (as P only features one branch); this augmented version is just the identity function from the extra input to the original output (remembering that the original input of the route was trivial). As for the route of node A, its augmented version features no extra input (as A features no bifurcation choice), and two extra outputs: the first one encodes whether branch A^0 happened, the second one encodes whether branch A^1 happened. This augmented version (which has two inputs and four outputs) is the partial function defined by $(0,0) \mapsto (0,0,1,0)$ and $(1,1) \mapsto (1,1,0,1)$ and undefined on inputs $(0,1)$ and $(1,0)$ (which are outside the practical input values). The same goes for B. Finally, the augmented version of F's route is identical to the original version (there are no bifurcation choices and only one branch).

As represented in Figure 14, the augmented version of a route features extra ancillary wires. One can then form a relation by connecting the non-ancillary wires of the λ_N^{aug}'s according to the indexed graph Γ (see Figure 15 for an example)[14]. We call this the 'choice

[14]Similarly to before, this procedure makes sense because relations form a traced monoidal category. The λ_N^{aug} are here viewed as relations, as any partial function can be.

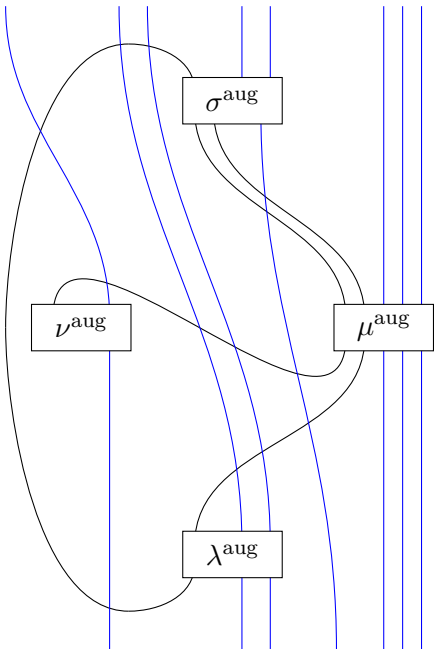


Figure 15: The 'choice relation' for the routed graph of Figure 12b. Here, we are assuming that ν has one branch, that λ and σ have two (with one of σ's branches having trivial bifurcation choices), and μ has three. For better readability, trivial wires are left implicit and ancillary wires are written in blue.

relation', which we will write $\Lambda_{(\Gamma,(\lambda_N)_N)}$. $\Lambda_{(\Gamma,(\lambda_N)_N)}$ goes from the bifurcation choices to the `Happens` binary variables that tell us whether each branch happened. The requirement that the former unambiguously determine the latter then takes a natural form.

Principle 1 (Univocality and bi-univocality)**.** *A routed graph* $(\Gamma,(\lambda_N)_N)$ *satisfies the principle of* <u>*univocality*</u> *if its choice relation* $\Lambda_{(\Gamma,(\lambda_N)_N)}$ *is a function.*

$(\Gamma,(\lambda_N)_N)$ *satisfies the principle of* <u>*bi-univocality*</u> *if both it and its adjoint* $(\Gamma^\top,(\lambda_N^\top)_N)$ *satisfy univocality.*

The adjoint of a routed graph is simply the routed graph obtained by reversing the direction of its arrows, and taking the adjoints of its routes: it can be interpreted as its time-reversed version. Being bi-univocal thus means being 'univocal both ways'.

When univocality is satisfied, the choice relation – which is then a choice function – plays another role: its causal structure (defined by functional dependence) tells us which bifurcation choices can affect the status of which branch. This will define the green dashed arrows in the branch graph, whereas the analogous information in the choice function of the adjoint graph will define the (reverse of the) red dashed arrows.

Our last job is to define the solid arrows in the branch graph. The idea is that the 'N^α' branch of node N has a direct influence on the 'M^β' branch of node M if there is an arrow from N to M that doesn't become either inconsistent or trivial (i.e. reduce to either zero sectors or to a single one-dimensional one) when one fixes N to be in branch α and M to be in branch β. To capture this, we will have to talk about *consistent assignments of values* to the indices of all arrows in the graph.

Definition 3.5 (Consistent assignment)**.** *A* <u>*consistent assignment*</u> *of values to* $(\Gamma,(\lambda_N)_N)$*'s indices is an assignment of a value to the arrows' indices, such that for any node* N*, the tuple of values for* N*'s inputs is related by* λ_N *to the tuple of values for* N*'s outputs.*

Note that (as proven in Appendix B) an assignment is consistent if and only if at every node, the tuple of input values and the tuple of output values that it yields are in the same branch (and in particular are not outside the practical inputs/outputs). In that sense, one can talk about this consistent assignment of values as, in particular, assigning a given branch to every node.

The idea of solid arrows, embodied by the following definition, is then that one draws a solid arrow from N^α to M^β if there is an arrow A joining N to M, except if N^α and M^β can never happen jointly, or if there is a single value of A's index compatible with both of them happening, and this value makes A trivial.

Definition 3.6. *Taking a branch N^α of node N and a branch M^β of node M, we say that there is a solid arrow $N^\alpha \to M^\beta$ if there exists an arrow from N to M, except if:*

- *there are no consistent assignment of values that assign branch α to N and branch β to M;*
- *or if all such assignments assign the same value to the index of the arrow $N \to M$, and this value has dimension 1 (i.e. corresponds to a one-dimensional sector).*

If there are several arrows from N to M, then we say that there is a solid arrow $N^\alpha \to M^\beta$ unless the above applies to all of them.

With this in our toolbox, we can define the branch graph.[15]

Definition 3.7 (Branch graph)**.** *The branch graph of a routed graph $(\Gamma, (\lambda_N)_N)$ that satisfies bi-univocality is the graph in which:*

- *the nodes are given by the branches of $(\Gamma, (\lambda_N)_N)$'s nodes;*
- *solid arrows are given by the previous definition;*
- *there is a green dashed arrow $N^\alpha \to M^\beta$ if the choice function $\Lambda_{(\Gamma,(\lambda_N)_N)}$ features causal influence (i.e. functional dependence) from $\texttt{Ind}^{\text{out}}_{N^\alpha}$ to $\texttt{Happens}_{M^\beta}$;*
- *there is a red dashed arrow $N^\alpha \to M^\beta$ if the choice function of the adjoint graph, $\Lambda_{(\Gamma^\top,(\lambda^\top_N)_N)}$, features causal influence (i.e. functional dependence) from $\texttt{Ind}^{\text{in}}_{M^\beta}$ to $\texttt{Happens}_{N^\alpha}$.*

Examples of branch graphs are shown in Figure 16. Now that the branch graph is defined, we can check whether it satisfies our second principle.

Principle 2 (Weak loops)**.** *We say that a loop in a branch graph is weak if it is entirely made of dashed arrows of the same colour.*

A routed graph satisfies the principle of weak loops if every loop in its branch graph is weak.

[15] Note that the branch graph is not defined if bi-univocality is not satisfied. Indeed, if either $\Lambda_{(\Gamma,(\lambda_N)_N)}$ or $\Lambda_{\left(\Gamma^\top,(\lambda^\top_N)_N\right)}$ is not a function but merely a relation, we cannot talk about its causal structure and therefore we cannot define either the green or the red dashed arrows. This has the important consequence that bi-univocality and weak loops are not logically independent principles; making sense of the second requires the first to hold.

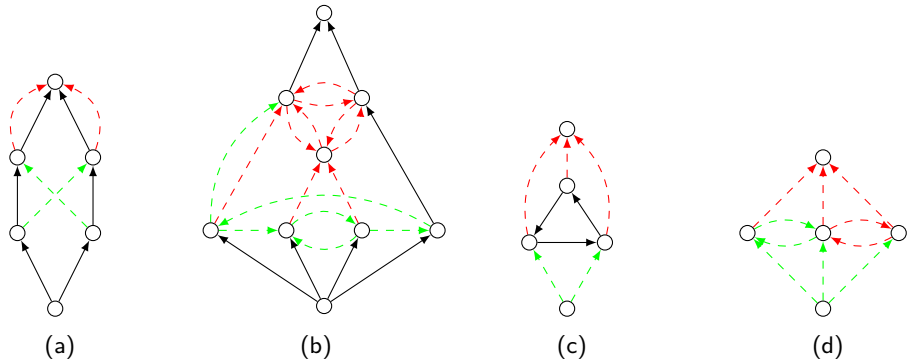


Figure 16: Examples of branch graphs. (a) and (b) satisfy the weak loops principle, but (c) and (d) do not. For (d), this is due to the presence of a bi-coloured ∞-shaped loop in the central layer.

Note that, as a particular case, any routed graph whose branch graph features no loops trivially satisfies this principle. This will be sufficient to check the consistency of processes featuring (possibly dynamical) coherent control of causal order. We will conjecture that the more exotic processes, which violate causal inequalities, are characterised by the existence of weak loops in their branch graph.

Finally, we can display our main theorem.

Theorem 3.1. *Let $(\Gamma, (\lambda_N)_N)$ be a routed graph satisfying the principles of bi-univocality and weak loops. (We then say that it is valid.) Then its associated skeletal supermap is a routed superunitary.*

The proof of Theorem 3.1 is given in Appendix B.

The next corollary, which is direct, stresses the fact that there are then many supermaps which can be obtained from this skeletal supermap, and that the validity of the latter implies that they are valid as well.

Corollary 3.1. *Let $(\Gamma, (\lambda_N)_N)$ be a valid routed graph. Then, any supermap built from its associated skeletal supermap by plugging in unitaries at some of its nodes and unitary monopartite supermaps at other nodes is a superunitary.*

4 Examples of constructing processes with indefinite causal order

In this section, we reconstruct three further examples of processes with indefinite causal order from valid routed graphs, namely the quantum 3-switch, the Grenoble process and the Lugano process. This will enable us to see each of these processes as a member of a large family of processes that can be constructed 'in the same way' – i.e. from the same routed graph. This in turn will allow us to distinguish between those features of the process that are 'accidental', and those that are essential for the consistency of the process.

What results is reminiscent of the situation for processes without indefinite causal order. Such processes can be represented as circuits, in which it is immediate that changing the particular transformations will preserve the consistency of the process, so long as the connectivity of the circuit is maintained. In our reconstructions of processes with indefinite causal order, it is immediate that changing the particular transformations in a routed circuit decomposition of a process with indefinite causal order preserves consistency, so long as the resulting routed circuit is still 'fleshing out' the same (valid) routed graph.

Before turning to the examples, we briefly explain a shorthand way of presenting the routed graphs. Rather than directly stating the route associated with each node, it is sometimes simpler to specify a *global index constraint*, from which the individual routes can be derived. This global index constraint specifies the allowed joint value-assignments for all of the indices in the graph. Formally, it can be represented as a Boolean tensor G over the Cartesian product of all the indices in the graph. We set the coefficients $G_i = 1$ to be equal to 1 for allowed joint value assignments i (note that i here denotes a list of indices), and $G_i = 0$ for those that are not allowed. Then, we can calculate the route λ_N at some specific node N as the least restrictive route consistent with the global index constraint. Assuming that the set of indices $\texttt{Ind}_N^{\text{in}}$ going into the node and the set of outgoing indices $\texttt{Ind}_N^{\text{out}}$ are disjoint, so that no arrows start and finish at that same node,[16] we can calculate this by marginalising over the indices $\texttt{Ind}\backslash\{\texttt{Ind}_N^{\text{in}} \sqcup \texttt{Ind}_N^{\text{out}}\}$ that do not come out of or go into the node. Writing the indices as $i = (i_{\text{in}} i_{\text{out}} i')$, where the $i' \in \{\texttt{Ind}_N^{\text{in}} \sqcup \texttt{Ind}_N^{\text{out}}\}$ denote the joint value-assignments of those 'irrelevant indices', the marginalisation is performed by taking the Boolean sum over i',

$$(\lambda_N)_{i_{\text{in}}}^{i_{\text{out}}} := \sum_{i'} G_{i_{\text{in}} i_{\text{out}} i'} \,. \tag{11}$$

In the examples below, we represent the global index by using a combination of index-matching on the routed graph and 'floating' equations relating the indices written beside the graph. The idea with index-matching is that when indices on two different arrows are matched, the global index constraint must be 0 for all joint value assignments in which they are not equal. Similarly, the global index constraint is 0 for all joint value assignments not satisfying the floating equations. The global index constraint of the routed graph is then the most general Boolean matrix compatible with the index-matching and the equations.

Similarly, we will also present routed *circuits* using a global index constraint. In that case, we derive the routes associated with the individual *transformations* (i.e. the boxes) that make up the circuit by marginalising over the global index constraint of the circuit.

We want to stress, once again, that the use of index-matching and global index constraints is only a graphical shorthand: in order to study the graphs and check the principles, they have to be formally translated into routes for the nodes.

4.1 The quantum 3-switch

The quantum 3-switch [14] is a unitary process defined analogously to the quantum switch, but with three intermediate agents: Alice (A), Bob (B) and Charlie (C). The Past (P) consists of a 6-dimensional control qudit P_C and a d-dimensional target qudit P_T. Depending on the initial state of the control qudit, the three agents receive the target qudit in a different order, outlined in Table 1. At the end, the target qudit is sent to the Future (F).

4.1.1 The routed graph

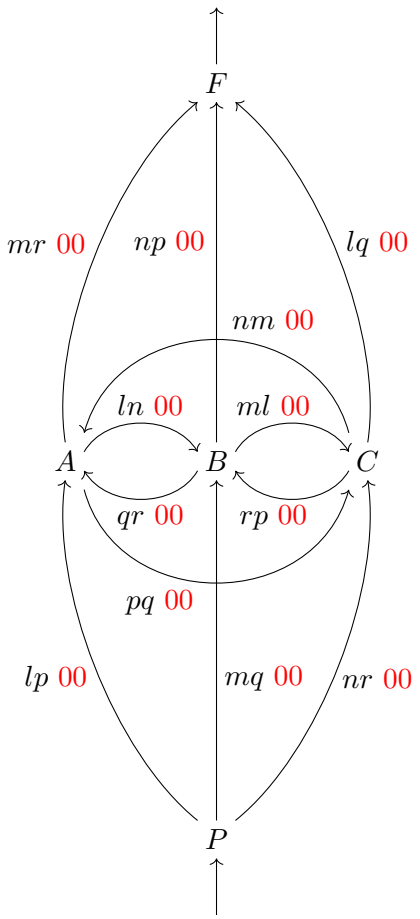
We start by drawing a routed graph from which the quantum 3-switch, amongst other processes, can be constructed. This routed graph is given in Figure 17. The global index

[16] Note that the requirements of bi-univocality and weak loops imply that any indices on these 'self-loops' would have to have values that either never happen, or else correspond to one-dimensional branches, and that the value of the indices are fixed by the other ingoing and outgoing arrows. This means that one gets exactly the same unitary processes from the routed graph if one removes the self-loops. Accordingly, the assumption of no self-loops does not sacrifice any generality.

Control state	Order
$\|1\rangle$	$A - B - C$
$\|2\rangle$	$A - C - B$
$\|3\rangle$	$B - C - A$
$\|4\rangle$	$B - A - C$
$\|5\rangle$	$C - A - B$
$\|6\rangle$	$C - B - A$

Table 1: The relative order of the agents Alice (A), Bob (B) and Charlie (C) depending on the value of the control state.

constraint is represented by matching the indices on different arrows, and by the floating equation $l + m + n + p + q + r = 1$. This equation enforces that precisely one of the six summed over indices is equal to one. Thus the global index constraint is the Boolean matrix that ensures that matched indices take the same value, and that exactly one of the six distinct values is 1.



$$l + m + n + p + q + r = 1$$

Figure 17: The routed graph for the quantum 3-switch, using a global index constraint.

The route at node P (which we denote η) is, by definition, the most liberal route compatible with the global index constraint. This is the route that forces exactly one of

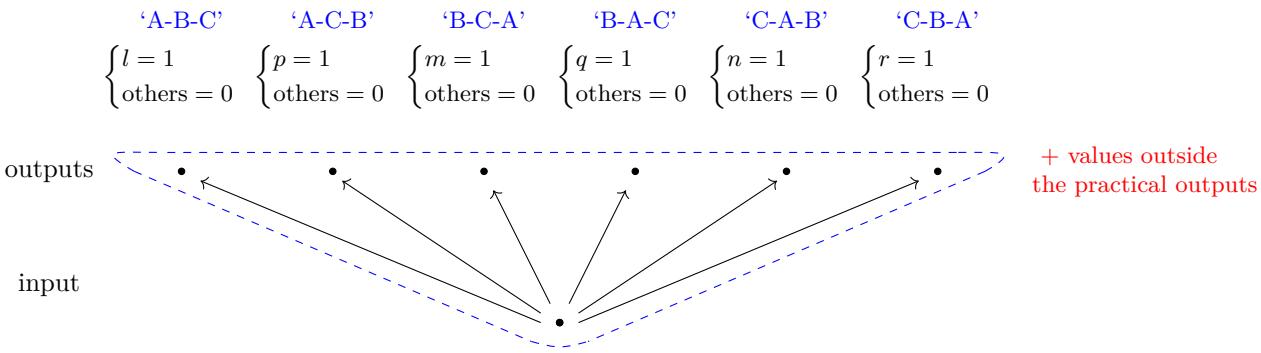


Figure 18: The route and branch structure of the node P of the quantum 3-switch. There is one unique branch with a bifurcation choice between six options, each of which enforces a causal order.

its indices to be equal to 1:

$$\begin{cases} \eta^{100000} = \eta^{010000} = \eta^{001000} = \eta^{000100} = \eta^{000010} = \eta^{000001} = 1\,; \\ \eta^{lmnpqr} = 0 \quad \text{otherwise.} \end{cases} \tag{12}$$

The route η also has a convenient graphical representation, depicted in Figure 18. η has a single branch with a bifurcation choice between six options, each corresponding to one of the indices $lmnpqr$ being equal to 1. Each option enforces one of the six possible causal orders.

Let us explain how this works in detail. In the routed graph for the standard switch, the arrow $P \to A$ came with two index values, corresponding to whether or not Alice received the message first. But for the 3-switch, if Alice does receive the message first, then there are two further possibilities: either she comes first and the causal order is clockwise ($A{-}B{-}C$), or she comes first and the order is anticlockwise ($A - C - B$). For this reason, the arrow from P to A has three index values overall. The sectors where she gets the message first correspond to $(l = 1, p = 0)$ and $(l = 0, p = 1)$; while $(l = 0, p = 0)$ corresponds to a one-dimensional 'dummy' sector. Likewise, all internal wires are associated with three sectors; two non-'dummy' sectors for when one of their indices equals one, and a 'dummy' sector for when both are equal to zero[17].

Now suppose that the agent at P makes the '$l = 1$' bifurcation choice, so that the message is sent to Alice. The global index constraint then enforces the route at the nodes A, B, C depicted in Figure 19. Thus Alice's route implies that she has no choice but to preserve the value of l, meaning that she must send the message along the arrow from A to B, since this is her only outgoing arrow that does not correspond to a dummy sector when $l = 1$. Then Bob similarly has no choice but to pass the message to Charlie, and finally Charlie is forced to send the message into the Future. The net result is that the message moves inexorably along the path $P \to A \to B \to C \to F$ of arrows decorated with an l index, giving the causal order $A - B - C$. Thus, if an agent at P makes the bifurcation choice that $l = 1$, they pick out this causal order.

Similarly, any option from the bifurcation choice enforces one of the six possible causal orders. In this sense, the bifurcation choice at P is a choice between causal orders, just as in the case of the original quantum switch. This state of affairs – that the causal

[17]Note that the sectors with both indices equal to 1, although formally present, are irrelevant: they correspond to impossible joint assignments of values.

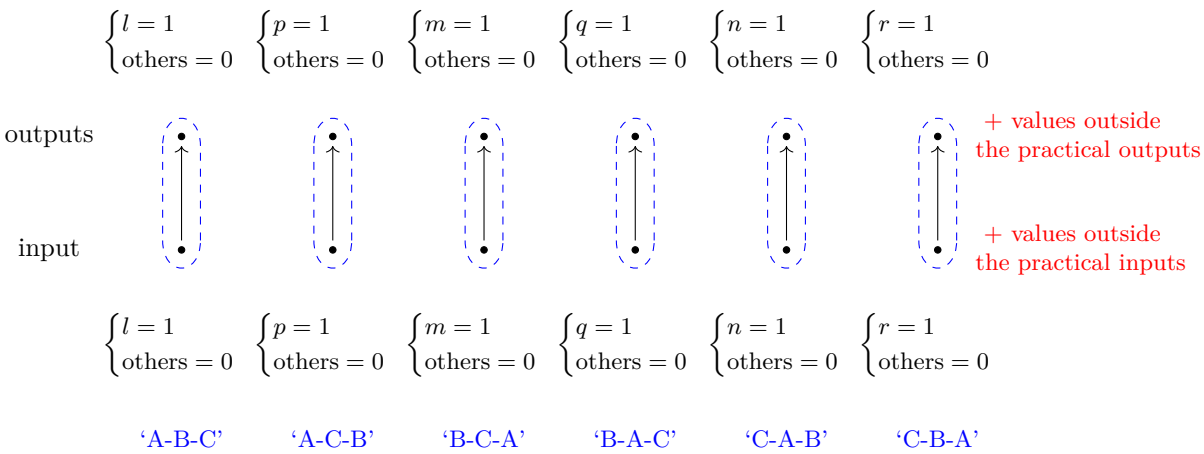


Figure 19: The route and branch structure for the intermediate agents in the 3-switch. There are six branches, each corresponding to a causal order.

order is determined by a bifurcation choice at the Past node – is characteristic of the (non-dynamical) coherent control of causal orders.

Now let us show that the routed graph satisfies our two principles. It is clear that the bifurcation choice at P, picking which index is equal to 1, determines the status of all branches of the intermediate nodes, since these branches are all defined by a certain index equalling 1 (see Figure 19). This bifurcation choice is the only one in the routed graph, and P and F each have just one branch (the route at F is just the time-reversed version of the one at P, obtained by reversing the direction of the arrows in Figure 18). Thus the sole bifurcation choice in the routed graph leads to a single branch happening at each node; formally speaking, we have a function from bifurcation choices to branch statuses. That is, the routed graph satisfies univocality.

Recall that bi-univocality requires that the time-reverse of the routed graph also satisfies univocality, where the time-reverse is obtained by reversing the arrows. Taking the time-reverse of Figure 17 and then relabelling

$$\begin{aligned} P &\longleftrightarrow F \\ l &\longleftrightarrow r \\ m &\longleftrightarrow p \\ n &\longleftrightarrow q \end{aligned} \tag{13}$$

results in exactly the same routed graph as Figure 17 itself. In other words, the routed graph is *time-symmetric* (up to relabelling). If a routed graph is univocal and time-symmetric, then it must also satisfy bi-univocality. Hence Figure 17 satisfies bi-univocality.

Since the routed graph satisfies bi-univocality, we can draw its branch graph, following the rules in Section 2.4: we display it in Figure 20. In this graph, the six branches for each of the nodes A, B and C are denoted by the specification of which index is equal to 1 (with all the others equal to 0), e.g. $A^{l=1}, A^{p=1}$, etc. There are no loops in the branch graph, meaning that the routed graph trivially satisfies weak loops. We can thus invoke Theorem 3.1 to conclude that any process that can be obtained from the routed graph of Figure 17, including the quantum 3-switch, is a consistent quantum process.

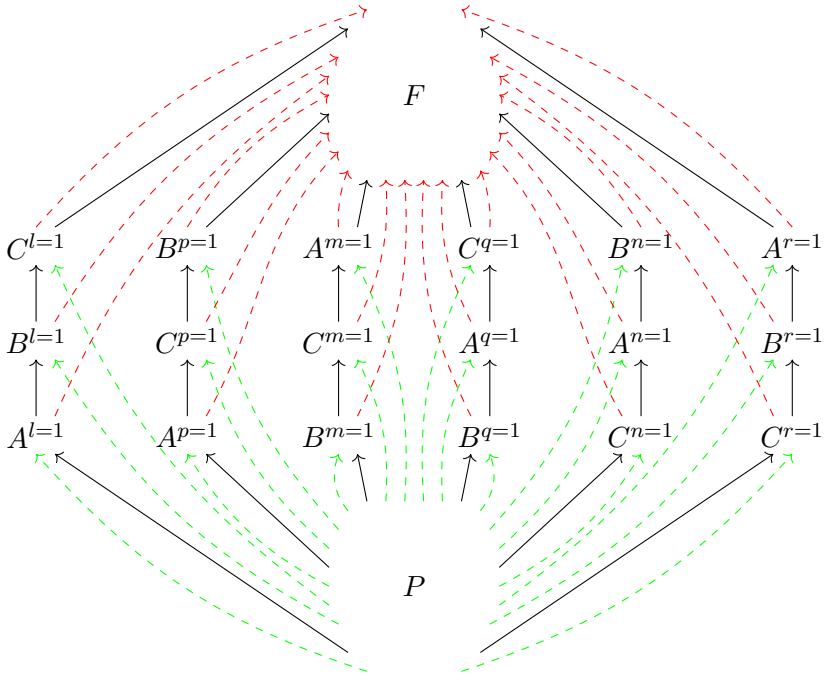


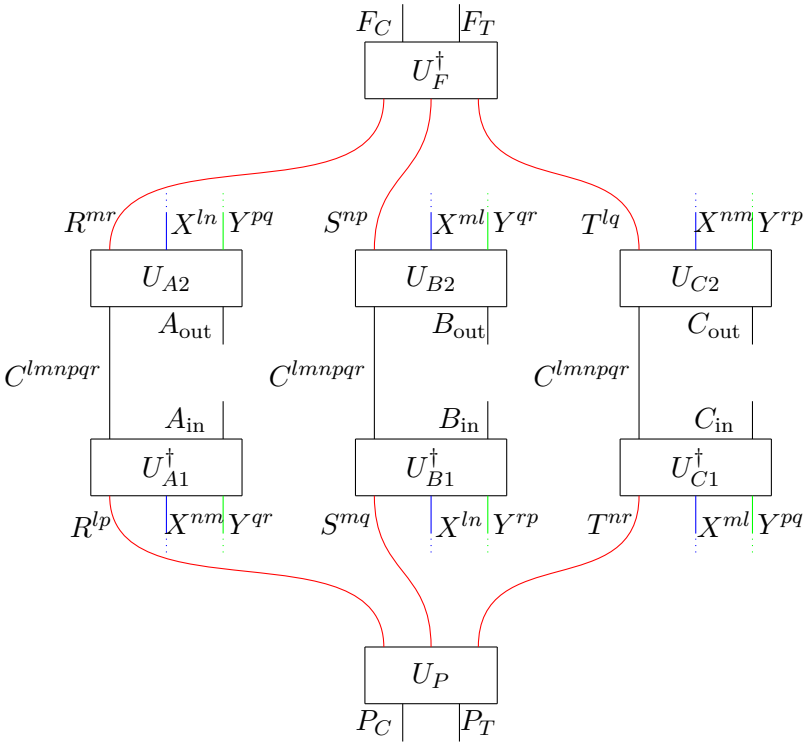
Figure 20: The branch graph for the quantum 3-switch.

4.1.2 The routed circuit

The routed circuit for the quantum 3-switch can be constructed from the routed graph in Figure 17 by inserting unitary transformations into the corresponding skeletal supermap. This is displayed in Figure 21, where we have again used the shorthand of global index constraints. The routes of the transformations can be derived from the global index constraint, just like the routes of the nodes in the routed graph.

The systems in the routed circuit have the following properties:

- The systems $P_T, F_T, A_{\text{in}}, A_{\text{out}}, B_{\text{in}}, B_{\text{out}}, C_{\text{in}}, C_{\text{out}}$ are all isomorphic, and correspond to a d-dimensional space.
- P_C, F_C are 6-dimensional control systems.
- The routed system C_k is also a 6-dimensional control system, with an explicit partition into six one-dimensional sectors.
- The routed systems $R^{lp}, S^{mq}, T^{nr}, X^{ln}, Y^{pq}$ are all $(2d+1)$-dimensional systems, this time partitioned into *two* d-dimensional sectors and a single 1-dimensional 'dummy' sector. For example, $R^{lp} = R^{00} \oplus R^{10} \oplus R^{01}$, where R^{00} is the 1-dimensional sector. The presence of two separate d-dimensional sectors corresponds to the fact that each of these wires can carry the message in two separate causal orders. We denote the unique state in the 1-dimensional sectors by $|\text{dum}\rangle$.



$$l + m + n + p + q + r = 1$$

Figure 21: A routed circuit diagram for the quantum 3-switch, using a global index constraint. To avoid too much clutter, instead of explicitly drawing loops, output lines that end with dots are to be interpreted as being looped back to join the corresponding input lines with dots (with the same system labels, including indices). Some wires are coloured for readability.

The unitary U_P at the bottom of the diagram is given by the isomorphism:

$$U_P : \begin{cases} |1\rangle_{P_C} \otimes |\psi\rangle_{P_T} \mapsto |\psi\rangle_{R^{10}} \otimes |\text{dum}\rangle_{S^{00}} \otimes |\text{dum}\rangle_{T^{00}} \\ |2\rangle_{P_C} \otimes |\psi\rangle_{P_T} \mapsto |\psi\rangle_{R^{01}} \otimes |\text{dum}\rangle_{S^{00}} \otimes |\text{dum}\rangle_{T^{00}} \\ |3\rangle_{P_C} \otimes |\psi\rangle_{P_T} \mapsto |\text{dum}\rangle_{R^{00}} \otimes |\psi\rangle_{S^{10}} \otimes |\text{dum}\rangle_{T^{00}} \\ |4\rangle_{P_C} \otimes |\psi\rangle_{P_T} \mapsto |\text{dum}\rangle_{R^{00}} \otimes |\psi\rangle_{S^{01}} \otimes |\text{dum}\rangle_{T^{00}} \\ |5\rangle_{P_C} \otimes |\psi\rangle_{P_T} \mapsto |\text{dum}\rangle_{R^{00}} \otimes |\text{dum}\rangle_{S^{00}} \otimes |\psi\rangle_{T^{10}} \\ |6\rangle_{P_C} \otimes |\psi\rangle_{P_T} \mapsto |\text{dum}\rangle_{R^{00}} \otimes |\text{dum}\rangle_{S^{00}} \otimes |\psi\rangle_{T^{01}} \end{cases} \tag{14}$$

between the non-routed system $P_C \otimes P_T$ ($6d$-dimensional) and the routed system $\bigoplus_{lmnpqr} \eta^{lmnpqr} R^{lp} \otimes S^{mq} \otimes T^{nr}$ [also of dimension $2(d\times1\times1)+2(1\times d\times1)+2(1\times1\times d) = 6d$].

U_F has the same form as U_P, where the $|1\rangle_{F_C}$ state of the control qubit is again mapped to the $l = 1$ sector, $|2\rangle_{F_C}$ is again mapped to $p = 1$ sector, and so on. The other unitaries denoted by U are the unique unitaries of the form above that respect the index-matching.

4.2 The Grenoble process

In their 2021 paper [13], Wechs and co-authors from Grenoble presented a new tripartite process with *dynamical* indefinite causal order, that is, where the causal order is not predetermined at the start of the process, but can be influenced by the intermediate agents themselves. In the present work, we shall call this process the *Grenoble process*.

Like the 3-switch, the Grenoble process involves three intermediate agents, who receive information from the global Past and ultimately send information into the global Future.[18] The Past (P) consists of a 3-dimensional control qutrit P_C and a 2-dimensional target qubit P_T. As with the previous processes we have studied, the logical state of the control system determines which of the intermediate agents will receive the message first. However, unlike the previous processes, this control system does not enforce a single causal order. This is because the agent who receives the message first is free to choose which agent will receive it second. In particular, the logical state of the target qubit after it passes through the first node will determine who gets it second: $|0\rangle$ means it will be sent in clockwise order (for example, to Bob if Alice was first), while $|1\rangle$ means it will be sent in anticlockwise order (for example, to Charlie if Alice was first). Finally, before the action of the third and final agent, the information about the relative order of the first two agents is scrambled onto an ancillary qubit, which is transferred directly to the Future (F).

In the Grenoble process, the emergent causal order depends not only on the global Past, but also on the actions of the intermediate agents. This is the hallmark of *dynamical* coherent control of causal order. In our terms, this will correspond to the fact that a causal order (and the branch statuses that fix it) is determined not only by a bifuraction choice at the Past, but also by bifurcation choices of the intermediate agents.

4.2.1 The routed graph

To begin with, we write down a routed graph from which the Grenoble process, amongst others, can be constructed. This graph is given in Figure 22, again using global index constraints.

For each arrow, the sector corresponding to all of its indices being equal to zero is a one-dimensional sector. The global index constraint (in particular, the floating equation $l + m + n = 1$), imposes a route at the node P that forces exactly one of the outgoing indices to equal 1, depicted in Figure 23. The route at the F node is just the time-reverse of the route at the Past. The global index constraint also gives rise to a route at A depicted in Figure 24. The routes at B and C are closely analogous.

Just like the routed graph for the 3-switch, the bifurcation choice at P determines which agent comes first. But unlike the 3-switch, this bifurcation does not enforce an entire causal order. Rather, it is left to the first intermediate agent to decide which one should come second. For example, suppose that an agent at P makes the bifurcation choice $l = 1$. This sends the message to Alice (since the only outgoing arrow from P that is not associated with a trivial dummy sector in this case is $P \to A$). This leads to Alice having the binary bifurcation choice associated with the branch $A^{l=1}$, depicted in Figure 24. This bifurcation choice determines which agent comes second. For example, suppose Alice chooses the bifurcation option $l_1 = 1$. Then the message is passed along the '$l_1 n_2$'-indexed arrow to Bob. Then Bob's route implies he has no such choice: he is forced to preserve the value of $l_1 = 1$, and is thereby compelled to send the message along the $m_1 l_1$ arrow to Charlie (he is confined to a branch $B^{l_1=1}$, analogous to $A^{n_1=1}$ in Figure 24). Finally, Charlie, confined to a branch $C^{h=1}$ analogous to $A^{f=1}$ in Figure 1, is forced to send the message off into the Future. Thus Alice's choice $l_1 = 1$ enforces the clockwise causal order $A - B - C$. On the other hand, choosing $l_2 = 1$ leads to the anticlockwise order $A - C - B$.

The situation is analogous if another one of the agents comes first. If Bob comes

[18] Note, that in the original formulation in Ref. [13], the Future is split into more than one party, whilst in this work, to simplify the presentation we consider only one Future party.

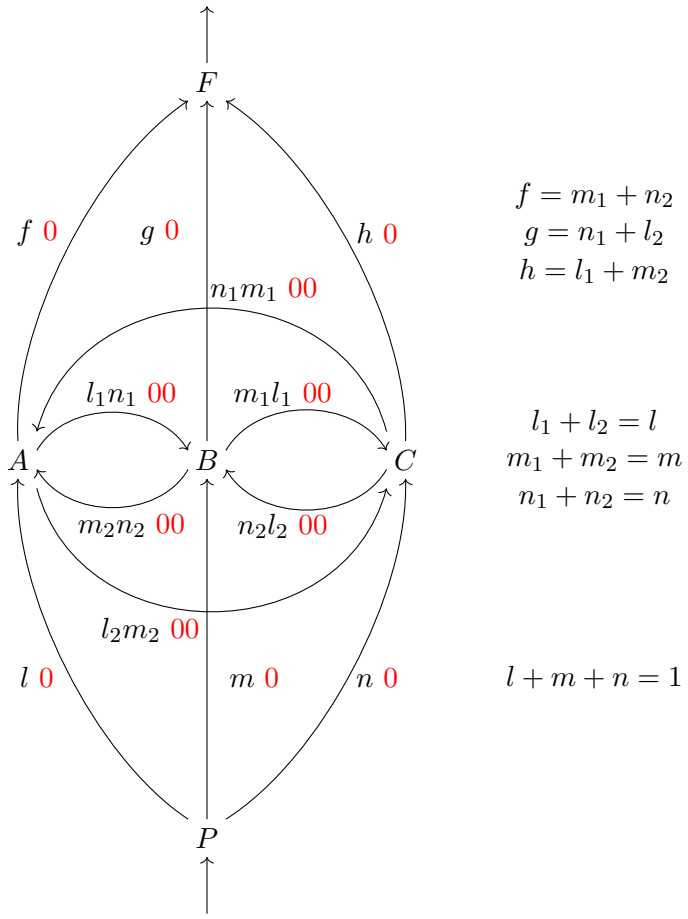


Figure 22: The routed graph for the Grenoble process, using a global index constraint.

first, he makes a bifurcation choice between $m_1 = 1$ and $m_2 = 1$ that enforces either the clockwise order $B - C - A$ or the anticlockiwse order $B - A - C$, respectively. Finally, if Charlie comes first, he chooses between $n_1 = 1$ and the clockwise order order $C - A - B$, or $n_2 = 1$ and the anticlockwise order $C - B - A$.

This scenario also allows for the potential disappearance of the information about the order of agents that acted already. Indeed, suppose that Alice comes last. This means she has either received the message coming clockwise from Charlie, or anticlockwise from Bob: i.e., either $m_1 = 1$ or $n_2 = 1$, respectively. In both cases, the floating equation $f = m_1 + n_2$ guarantees that $f = 1$, meaning that the information about which agent came first and which came second may be deleted.[19] This can be seen in the structure of

[19]This does not mean, however, that this information is *necessarily* lost; Alice might plug in an operation that carries it into the future. The point is that that the route structure makes no requirement for this information to be preserved in a specific structural way. Alice might therefore also plug in an operation that unitarily rotates, degrades, or outright discards this information.

This point may be better understood by appealing to the symmetry between this reverse, 'backward-facing' bifurcation (visible in Figure 24 ,in the $A^{f=1}$ branch in which Alice comes last), and a 'forward-facing' bifurcation (as present e.g. in the branch $A^{l=1}$ in which Alice comes first). In the latter, we colloquially say that Alice 'chooses' who comes second, which she could do e.g. by controlling her output state on an ancilla; but she might as well just plug in a unitary, so that this information corresponds to some information present in the state of her main input, coming from the past. Either way, what the route structure indicates is that *from this point on, this information is tracked by the route structure.* The situation with a backwards-facing bifurcation, as present in the $A^{f=1}$ branch, is to be understood

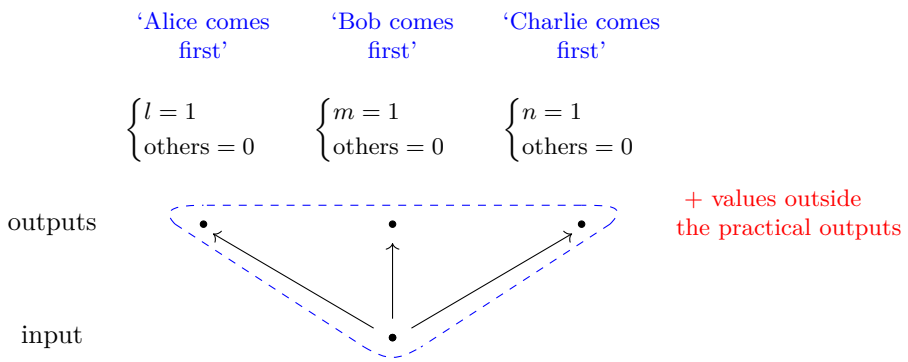


Figure 23: The branch structure for the node P of the Grenoble process. The bifurcation choice in the unique branch determines which agent comes first.

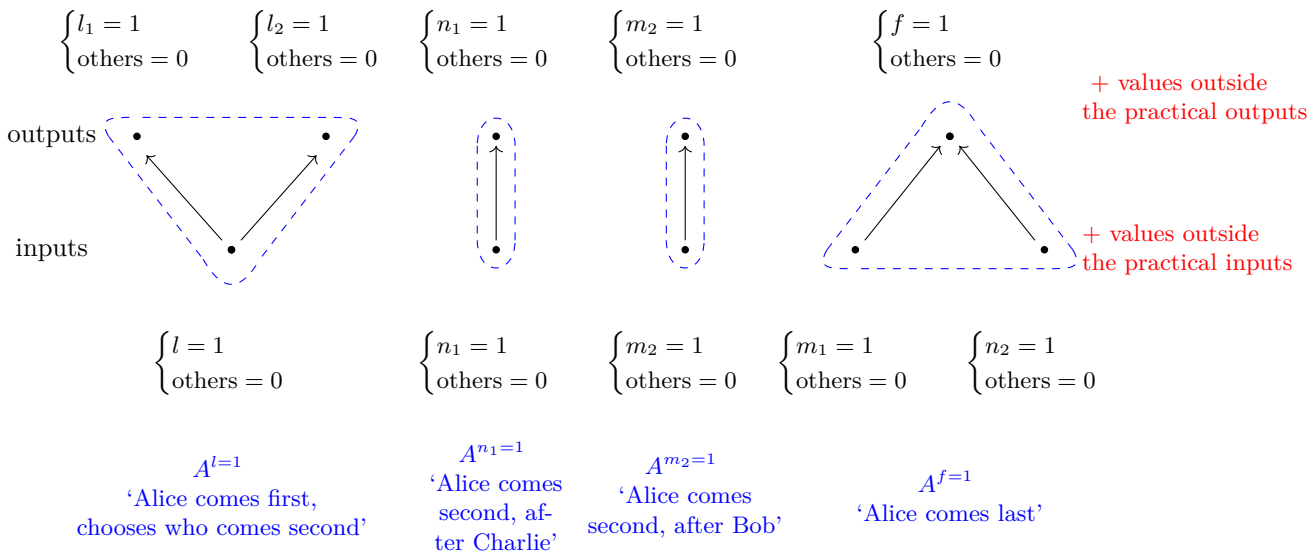


Figure 24: The branch structure for the node A of the Grenoble process.

the '$f = 1$' branch in Figure 24. Again, the situation is analogous if another agent comes last.

To construct the branch graph, consider the following. The node P consists of a single branch with a bifurcation choice between three options, each corresponding to the case when one of the three indices l, m, n equals 1. In the time-reversed version of the routed graph, the node F has a bifurcation with three options, each corresponding to the case when one of the three indices f, g, h equals 1.

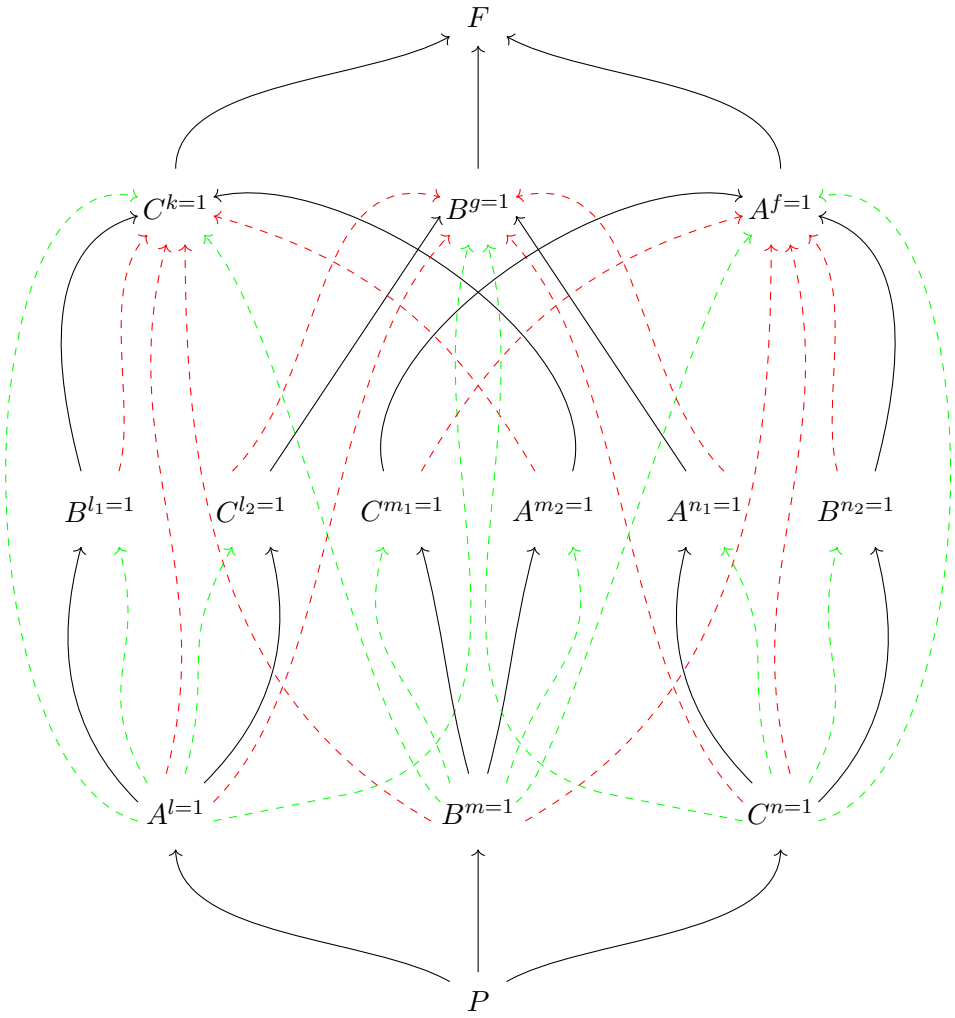


Figure 25: The branch graph for the Grenoble process. For clarity, we have omitted the green and red dashed arrows from P and to F, respectively: they simply point upwards in the diagram to/from all of the other branched nodes.

The routes at the nodes A, B, C consist of four branches, as illustrated in Figure 24. One of these branches corresponds to the case when the index on the wire coming directly from P equals 1, with a bifurcation between two options splitting this index into an index of the same name with subscript either 1 or 2 (corresponding to whether the message is sent clockwise or anticlockwise). Another branch corresponds to the case when the index on the wire going directly to F equals 1, with a bifurcation in the time-reversed routed graph combining the second index of each of the two wires coming in from the other

symmetrically: from that point on, the information about who came first might be processed in an arbitrary way (including being deleted) by Alice or by later agents, depending on their choices of operation.

agents (corresponding to whether the message came from the clockwise or anticlockwise direction). The final two branches correspond to the cases when one of the two indices that appear on both the input and output wires of the nodes are equal to 1. Following Figure 24, we shall denote these branches by superscripts labelling which index is equal to 1.

Like the one for the 3-switch, the routed graph for the Grenoble process is time symmetric. To see this, note that reversing the direction of the arrows and relabelling

$$\begin{aligned} P &\longleftrightarrow F \\ l &\longleftrightarrow f \\ m &\longleftrightarrow g \\ n &\longleftrightarrow h \\ l_1 &\longleftrightarrow n_2 \\ l_2 &\longleftrightarrow m_1 \\ m_2 &\longleftrightarrow n_1 \end{aligned} \tag{15}$$

results in exactly the same graph.

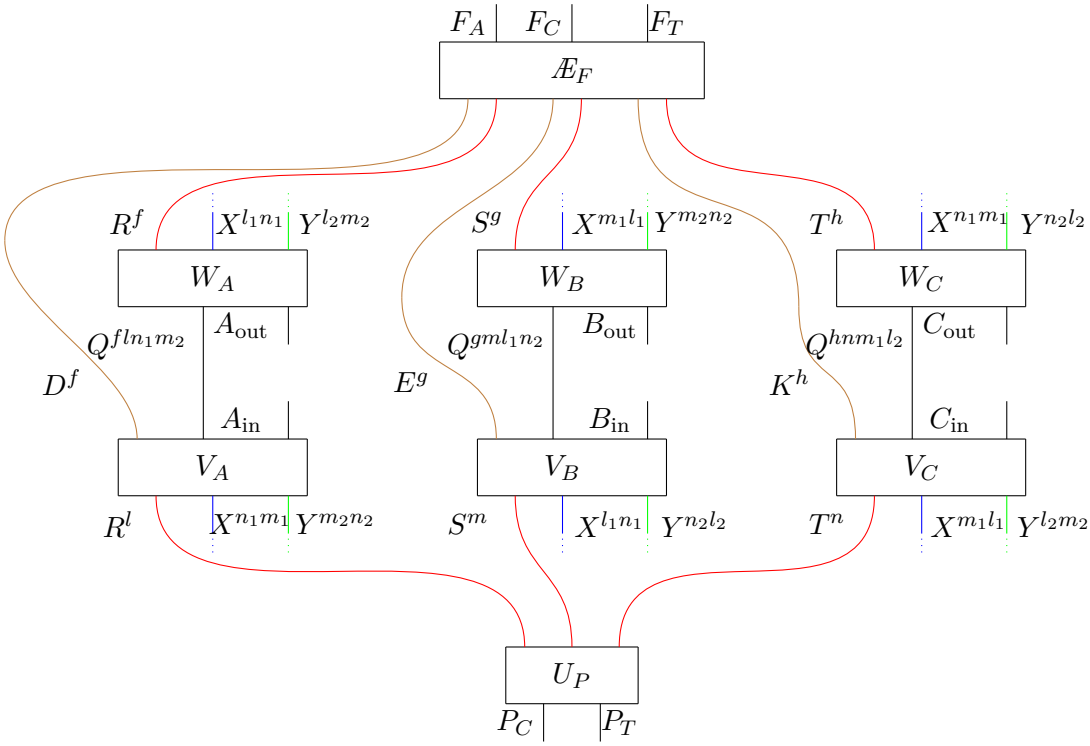
The fact that the routed graph satisfies univocality is implicit in the above explanation of how the bifurcation choices pick out a causal order. For they do so precisely by determining which branch happens at each node. For example, the bifurcation choice $l = 1$ at P leads to the branch $A^{l=1}$ happening, corresponding to Alice coming first. Then Alice's bifurcation choice determines which branches happen at B and C: $B^{l_1=1}$ and $C^{h=1}$ if she chooses $l_1 = 1$; $C^{l_2=1}$ and $B^{g=1}$ if she chooses $l_2 = 1$. In general, the bifurcation choices at P and at the resulting first intermediate node always determine which branches happen. Thus, the routed graph satisfies univocality. Since it is time-symmetric, it immediately follows that it also satisfies bi-univocality.

This allows us to draw the branch graph, which is shown in Figure 25. Since this branch graph has no loops, it trivially satisfies our weak loops condition. Thus the routed graph is valid. It follows that the Grenoble process – and any other process constructed from this routed graph – is consistent.

4.2.2 The routed circuit

By inserting suitable unitary transformations into the skeletal supermap associated with Figure 22, we can now construct the Grenoble process. A routed circuit for the Grenoble process is given in Figure 26:

- The systems $P_T, F_T, A_{\text{in}}, A_{\text{out}}, B_{\text{in}}, B_{\text{out}}, C_{\text{in}}, C_{\text{out}}$ are all isomorphic, and correspond to a 2-dimensional target Hilbert space (encoding the message).
- P_C is a 3-dimensional control system; F_C is a 3-dimensional control system, F_A is a 2-dimensional ancillary system.
- The routed systems $Q^{fln_1m_2}, Q^{gml_1n_2}, Q^{hnm_1l_2}$ are 4-dimensional control systems, with an explicit partition into four 1-dimensional sectors, each corresponding to exactly one of their four indices being equal to one.
- The routed systems $R^l, R^f, S^m, S^g, T^n, T^h$ are 3-dimensional systems, with an explicit partition into one 1-dimensional 'dummy' sector and one 2-dimensional 'message' sector, for example, $R^l = R^0 \oplus R^1$, where R^0 is the 1-dimensional sector.



$$l + m + n = 1$$

$$l_1 + l_2 = l$$
$$m_1 + m_2 = m$$
$$n_1 + n_2 = n$$

$$f = m_1 + n_2$$
$$g = n_1 + l_2$$
$$h = l_1 + m_2$$

Figure 26: A routed circuit diagram for the Grenoble process, using a global index constraint. To avoid graphical clutter, we have avoided explicitly drawing loops. Instead, output lines that end with dots are to be interpreted as being looped back to join the corresponding input lines with dots (with the same system labels, including indices). Wires are coloured for better readability.

- The routed systems D^f, E^g, K^h are similarly 3-dimensional systems, with an explicit partition into one 1-dimensional 'dummy' sector and one 2-dimensional 'ancillary' sector, for example, $D^l = D^0 \oplus D^1$, where D^0 is the 1-dimensional sector. The 2-dimensional 'ancillary' system will be used to store the information about whether the message was sent clockwise or anticlockwise (or in a superposition of the two) after the first agent, conditional on the state of the qubit before the action of the third agent.

- The routed systems $X^{n_1m_1}, Y^{m_2n_2}, X^{l_1n_1}, Y^{n_2l_1}, X^{m_1l_1}, Y^{l_2m_2}$ are all 4-dimensional systems, partitioned into *one* 2-dimensional 'message' sector (corresponding to the message travelling from the second to the third agent), *one* 1-dimensional 'message' sector (corresponding to the message travelling from the first to the second agent, in which case the space is only one-dimensional because the state of the message itself determines to whom it is sent next), and *one* 1-dimensional 'dummy' sector. For example, $X^{n_1m_1} = X^{00} \oplus X^{10} \oplus X^{01}$, where X^{00} is the 1-dimensional 'dummy' sector and X^{10} is the 1-dimensional 'message' sector.

The global index imposes a route $\delta^{(l+m+n),1}$ on U_P that forces exactly one of its output

indices to be equal to 1. In other words, its practical output space is $\bigoplus_{lmn} \delta^{(l+m+n),1} R^l \otimes S^m \otimes T^n$.

U_P is a three-party generalisation of the superposition-of-paths unitary (3) from Section 2. Its action is given by the following, where we label the kets by individual sectors, rather than by systems:

$$U_P : \begin{cases} |1\rangle_{P_C} \otimes |\psi\rangle_{P_T} \mapsto |\psi\rangle_{R^1} \otimes |\mathrm{dum}\rangle_{S^0} \otimes |\mathrm{dum}\rangle_{T^0} \\ |2\rangle_{P_C} \otimes |\psi\rangle_{P_T} \mapsto |\mathrm{dum}\rangle_{R^0} \otimes |\psi\rangle_{S^1} \otimes |\mathrm{dum}\rangle_{T^0} \\ |3\rangle_{P_C} \otimes |\psi\rangle_{P_T} \mapsto |\mathrm{dum}\rangle_{R^0} \otimes |\mathrm{dum}\rangle_{S^0} \otimes |\psi\rangle_{T^1} \end{cases} \tag{16}$$

Thus U_P defines a unitary transformation from $P_C \otimes P_T$ to $\bigoplus_{lmn} \delta^{(l+m+n),1} R^l \otimes S^m \otimes T^n$. In fact, the global index constraint (in particular, the floating equation $l + m + n = 1$) restricts U_P's practical output space to $\bigoplus_{lmn} \delta^{(l+m+n),1} R^l \otimes S^m \otimes T^n$, meaning that it defines a routed unitary transformation.

V_A is defined below. Note that here the labelling by sectors is necessary to distinguish between states belonging to different sectors that we label with the same ket, e.g. $|0\rangle_{X^{01}}$ and $|0\rangle_{X^{10}}$.

$$V_A : \begin{cases} |0\rangle_{R^1} \otimes |\mathrm{dum}\rangle_{X^{00}} \otimes |\mathrm{dum}\rangle_{Y^{00}} \mapsto |0\rangle_{A_{\mathrm{in}}} \otimes |0\rangle_Q \otimes |\mathrm{dum}\rangle_{D^0} \\ |1\rangle_{R^1} \otimes |\mathrm{dum}\rangle_{X^{00}} \otimes |\mathrm{dum}\rangle_{Y^{00}} \mapsto |1\rangle_{A_{\mathrm{in}}} \otimes |0\rangle_Q \otimes |\mathrm{dum}\rangle_{D^0} \\ |\mathrm{dum}\rangle_{R^0} \otimes |0\rangle_{X^{10}} \otimes |\mathrm{dum}\rangle_{Y^{00}} \mapsto |0\rangle_{A_{\mathrm{in}}} \otimes |1\rangle_Q \otimes |\mathrm{dum}\rangle_{D^0} \\ |\mathrm{dum}\rangle_{R^0} \otimes |\mathrm{dum}\rangle_{X^{00}} \otimes |1\rangle_{Y^{10}} \mapsto |1\rangle_{A_{\mathrm{in}}} \otimes |2\rangle_Q \otimes |\mathrm{dum}\rangle_{D^0} \\ |\mathrm{dum}\rangle_{R^0} \otimes |0\rangle_{X^{01}} \otimes |\mathrm{dum}\rangle_{Y^{00}} \mapsto |0\rangle_{A_{\mathrm{in}}} \otimes |3\rangle_Q \otimes |0\rangle_{D^1} \\ |\mathrm{dum}\rangle_{R^0} \otimes |1\rangle_{X^{01}} \otimes |\mathrm{dum}\rangle_{Y^{00}} \mapsto |1\rangle_{A_{\mathrm{in}}} \otimes |3\rangle_Q \otimes |1\rangle_{D^1} \\ |\mathrm{dum}\rangle_{R^0} \otimes |\mathrm{dum}\rangle_{X^{00}} \otimes |0\rangle_{Y^{01}} \mapsto |0\rangle_{A_{\mathrm{in}}} \otimes |3\rangle_Q \otimes |1\rangle_{D^1} \\ |\mathrm{dum}\rangle_{R^0} \otimes |\mathrm{dum}\rangle_{X^{00}} \otimes |1\rangle_{Y^{01}} \mapsto |1\rangle_{A_{\mathrm{in}}} \otimes |3\rangle_Q \otimes |0\rangle_{D^1} \end{cases} \tag{17}$$

Since the global index constraint restricts V_A's practical input and output spaces to those sectors where exactly one index is equal to 1, it also defines a routed isometry[20]. V_B and V_C are defined similarly.

The routed unitary W_A is defined as follows:

$$W_A : \begin{cases} |0\rangle_{A_{\mathrm{out}}} \otimes |0\rangle_Q \mapsto |\mathrm{dum}\rangle_{R^0} \otimes |0\rangle_{X^{10}} \otimes |\mathrm{dum}\rangle_{Y^{00}} \\ |1\rangle_{A_{\mathrm{out}}} \otimes |0\rangle_Q \mapsto |\mathrm{dum}\rangle_{R^0} \otimes |\mathrm{dum}\rangle_{X^{00}} \otimes |1\rangle_{Y^{10}} \\ |0\rangle_{A_{\mathrm{out}}} \otimes |1\rangle_Q \mapsto |\mathrm{dum}\rangle_{R^0} \otimes |0\rangle_{X^{01}} \otimes |\mathrm{dum}\rangle_{Y^{00}} \\ |1\rangle_{A_{\mathrm{out}}} \otimes |1\rangle_Q \mapsto |\mathrm{dum}\rangle_{R^0} \otimes |1\rangle_{X^{01}} \otimes |\mathrm{dum}\rangle_{Y^{00}} \\ |0\rangle_{A_{\mathrm{out}}} \otimes |2\rangle_Q \mapsto |\mathrm{dum}\rangle_{R^0} \otimes |\mathrm{dum}\rangle_{X^{00}} \otimes |0\rangle_{Y^{01}} \\ |1\rangle_{A_{\mathrm{out}}} \otimes |2\rangle_Q \mapsto |\mathrm{dum}\rangle_{R^0} \otimes |\mathrm{dum}\rangle_{X^{00}} \otimes |1\rangle_{Y^{01}} \\ |0\rangle_{A_{\mathrm{out}}} \otimes |3\rangle_Q \mapsto |0\rangle_{R^1} \otimes |\mathrm{dum}\rangle_{X^{00}} \otimes |\mathrm{dum}\rangle_{Y^{00}} \\ |0\rangle_{A_{\mathrm{out}}} \otimes |3\rangle_Q \mapsto |1\rangle_{R^1} \otimes |\mathrm{dum}\rangle_{X^{00}} \otimes |\mathrm{dum}\rangle_{Y^{00}} \end{cases} \tag{18}$$

The routed unitaries W_B and W_C are defined in a similar way. Finally, the routed unitary

[20]The notion of a routed isometry is defined similarly to that of a routed unitary [16].

Æ is given by the following:

$$Æ_F : \begin{cases} |\psi\rangle_{R^1} \otimes |\xi\rangle_{D^1} \otimes |\mathrm{dum}\rangle_{S^0} \otimes |\mathrm{dum}\rangle_{E^0} \otimes |\mathrm{dum}\rangle_{T^0} \otimes |\mathrm{dum}\rangle_{K^0} \mapsto |\xi\rangle_{F_A} \otimes |1\rangle_{F_C} \otimes |\psi\rangle_{F_T} \\ |\mathrm{dum}\rangle_{R^0} \otimes |\mathrm{dum}\rangle_{D^0} \otimes |\psi\rangle_{S^1} \otimes |\xi\rangle_{E^1} \otimes |\mathrm{dum}\rangle_{T^0} \otimes |\mathrm{dum}\rangle_{K^0} \mapsto |\xi\rangle_{F_A} \otimes |2\rangle_{F_C} \otimes |\psi\rangle_{F_T} \\ |\mathrm{dum}\rangle_{R^0} \otimes |\mathrm{dum}\rangle_{D^0} \otimes |\mathrm{dum}\rangle_{S^0} \otimes |\mathrm{dum}\rangle_{E^0} \otimes |\psi\rangle_{T^1} \otimes |\xi\rangle_{K^1} \mapsto |\xi\rangle_{F_A} \otimes |3\rangle_{F_C} \otimes |\psi\rangle_{F_T} \end{cases} \tag{19}$$

Note that the Grenoble process is an isometric process, with the overall output dimension greater than the overall input dimension (in particular, V is a routed isometry). The process can be made unitary in a natural way, by adding an extra 2-dimensional ancillary qubit to the input of the Past and adding routed wires of dimension $1+2$ from the Past to each of the routed unitaries W, bearing the same index as the wire from the Past to the corresponding V. This makes the process symmetric in time. As a result, this increases the dimension of the Hilbert space of the sector carrying the message between the first and second agents from 1 to 2. In turn, this increases the dimensionality of the input space to the unitaries V, making the entire process unitary.

Note also that the Future cannot necessarily determine the relative order of the first two agents from their control and ancillary qubits F_C, F_A, if the third agent performs a non-unitary operation (because the order information encoded in the ancillary qubit relied on knowledge of the state of the message before the action of the third agent).

One peculiar feature of the Grenoble process is that the qubit that we have called the 'target qubit' – that is, the system that passes between the intermediate agents – plays a dual role. On the one hand, it is the 'message' that the agents receive. On the other hand, it also plays a role in determining the causal order. In particular, after it passes through the first agent, its logical state determines which agent receives it next. Thus if Alice comes first and wants to send the target qubit to Bob, she must send him the $|0\rangle$ state, but if she wants to send it to Charlie, she must send him $|1\rangle$.

Our reconstruction of the Grenoble process makes it obvious that this feature is not necessary to make the process consistent. Starting from the same routed graph, one can easily define a variation on the Grenoble process, in which Alice is also given a second, 'control' qubit. This control qubit determines which agent comes second, leaving Alice free to send that agent whatever state on the target qubit she likes. Bob and Charlie can also be given their own qubits. Since this process can be obtained by fleshing out a routed graph whose validity we have already checked, it is immediate that this new process is also consistent. This illustrates a useful feature of our framework for constructing processes with indefinite causal order; namely, that variations on a process can be defined in a straightforward way, leading to a clearer understanding of which features of the original process were essential for its logical consistency, and which other features can be changed at will.

4.3 The Lugano process

The Lugano process, discovered by Araújo and Feix and then published and further studied by Baumeler and Wolf [11, 12] (and therefore sometimes also called BW or AF/BW), is the seminal example of a unitary process violating causal inequalities. It was first presented as a classical process, whose unitary extension to quantum theory can be derived in a straightforward way [15]. As we place ourselves in a general quantum framework here, we will primarily focus on this quantum version of the process; we note that the classical version can be obtained from the quantum one by feeding it specific input states and introducing decoherence in each of the wires of the circuit. This shows, more generally,

that at least some exotic classical processes are also part of the class of processes that can be built through our procedure.

Indeed, we will show here how the (unitary) Lugano process can be constructed from a valid routed graph; this will provide an example of a process violating causal inequalities that can also be accommodated by our framework. In fact, we will derive a larger family of processes, defined by a same valid routed graph, and display how the Lugano process can be obtained as the simplest instance of this family. The other processes in this family share the basic behaviour of the Lugano process, but can feature, on top of it, arbitrarily large dimensions and arbitrarily complex operations.

4.3.1 The logical structure

Before we present the routed graph for the construction of the Lugano process, let us start with an intuitive account of the logical structure lying at the heart of it. This logical structure can be presented as a voting protocol involving three agents, in which each of the agents receives part of the result of the vote before having even cast their vote. Why this is possible without leading to any logical paradox, of the grandfather type, is the central point to understand.

In this voting protocol, each agent casts a vote for which of the other two agents they would like to see come last in the causal order. Alice, for instance, can either vote for Bob or for Charlie to come last. If there is a majority, then the winning agent can both i) learn that they won the vote, and ii) receive (arbitrarily large) messages from each of the two losing agents. As for the two losers, each of them can only learn that they lost (i.e. no majority was obtained in favour of them), and they cannot receive any messages from the other agents. If no majority is obtained, then all agents learn that they lost, and none of them can signal to any other.

This voting protocol would have nothing surprising if it assumed that the winner learns of their victory and receives messages from the losers 'after' all the votes are cast. Yet in the Lugano process, the crucial fact is that Alice, for instance, learns whether she won, and (if she won) receives Bob and Charlie's messages, *before* she casts her own vote; and the same goes for Bob and Charlie. This sounds dangerously close to a grandfather paradox, since each agent contributes to an outcome that they might become aware of before they make their contribution. It seems likely that the agents could somehow take advantage of this system to send messages back to their own past, and decide what they do based on those messages, leading to logical inconsistencies.

Why this never happens – why, more precisely, the agents still have no way to send information back to themselves – can be figured out with a bit of analysis of the voting system. Indeed, Alice, for instance, finds herself in either of two cases. The first one is that she won: a majority 'was' obtained in favour of her. This means that she cannot send messages to either of the agents, since only the winner can receive messages. Nor can she signal to other agents by casting her vote: her victory implies that both Bob and Charlie voted for her, in which case her own vote is irrelevant to the outcome. Therefore, if Alice wins, then she cannot send any information back to herself via the other agents.

Alternatively, Alice could lose the vote. If so, then she cannot receive any messages from the other agents, so she has no hope of sending information back to herself through their messages to her. Therefore, if she wants to send information to herself, she will have to try to change the outcome of the vote in her favour (thus creating a grandfather-type paradox). But she cannot do this by simply changing her own vote, as there being a majority in her favour only depends on how the other agents vote. Nor can she make herself win by encouraging the other agents to vote differently. Alice can only send a

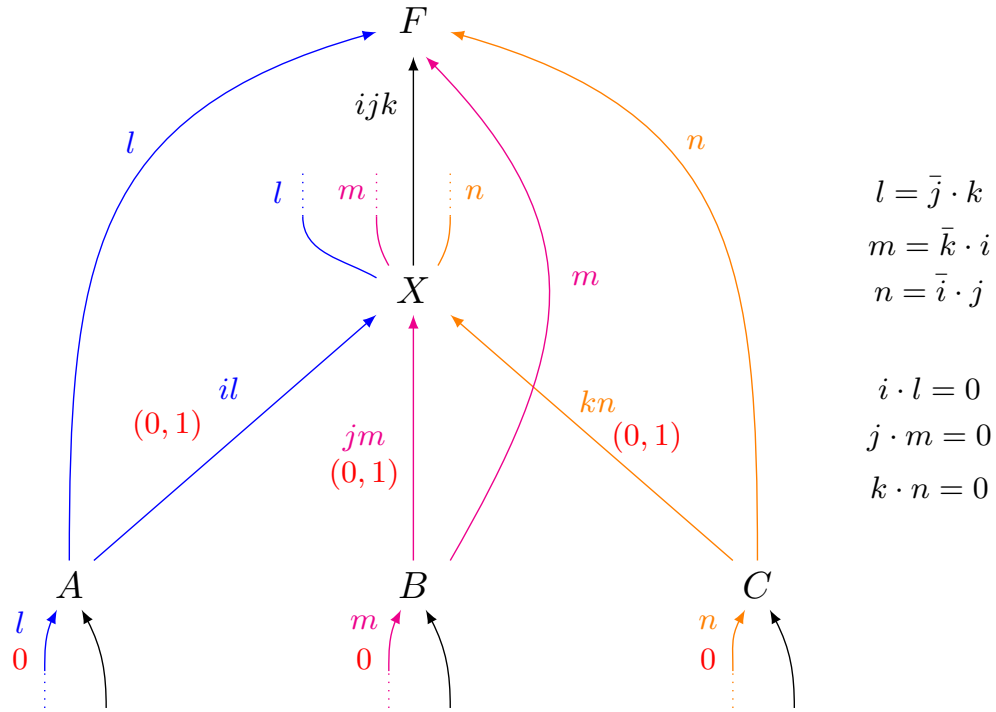


Figure 27: The routed graph for the Lugano process. To help intuition, we used different colours to denote the arrows that pertain to some particular agent (i.e. the ones whose indices encode the 'votes' or the 'vote result' for that agent). Each of the indices has only two possible values, 0 or 1. To reduce clutter, we have used arrows with dotted ends to avoid drawing all the arrows explicitly; pairs of dotted arrows with the same index are shorthand for a single unbroken arrow. Conditions in red next to an arrow indicate which sectors of that arrow have to be trivial; for instance, in the blue arrow from A to X, the sector $(i = 0, l = 1)$ is trivial.

message to (say) Bob if Charlie voted for Bob as well. So, whatever Bob does, there will never be a majority in favour of Alice. Therefore, if Alice loses, she cannot send any information back to herself. For this reason, the Lugano process, despite conflicting with intuitions about causal and temporal structure, does not lead to any logical paradoxes, after all.

4.3.2 The routed graph

Of course, our description of the Lugano process so far has only been pitched at an intuitive level. The point of the routed graph that we will now present is precisely to formalise this intuitive description; while the validity of this graph – defined as the satisfaction of our two principles – will provide a formal counterpart to our argument that no logical paradoxes should arise from this protocol. Our routed graph is depicted in Figure 27.

In this routed graph, the nodes A, B, and C, representing the three agents, are supplemented with an additional node X that can be thought of – continuing with our metaphor – as a 'vote-counting station', in which the votes for each of the agents are centralised and counted.[21]

[21] We stress that the inclusion of this auxiliary node is an *ad hoc* procedure, since our framework provides no way to infer the routed circuit decomposition of a given process. An informal intuition for this inclusion is that, when framed as a skeletal supermap, the Lugano process seems to require a node at which the information is processed and dispatched to other nodes. As we elaborate upon in the conclusion, deriving formally the routed circuit decomposition of a given process is an important subject for future research.

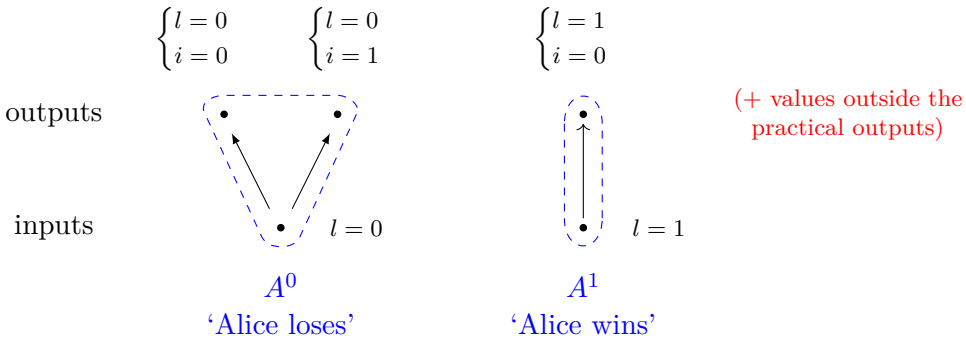


Figure 28: The branch structure of the route for node A. B and C are fully analogous.

Let us determine the routes in the graph and explain their meaning, starting with node A. The index l of the arrow going into A from its vote-counting station X indicates whether Alice won the vote (it has value 1 if she wins, and 0 if she loses). The index i encodes whether Alice votes for Bob ($i = 1$) or Charlie ($i = 0$). The constraint $i \cdot l = 0$ enforces that, if Alice wins, her vote (which is then irrelevant anyway) is recorded as a vote for Charlie.[22] This leads to the route for A depicted in Figure 28. The route consists of two branches corresponding to Alice's victory or defeat, with a binary bifurcation choice representing her own vote in the losing branch. The routes for B and C are fully analogous.

Now let us consider X, the vote-counting station. This node features all of the graph's indices in both its inputs and its outputs. The values of l, m, n are uniquely determined by those of i, j, k through the global index constraints, and it can be checked that there are 5 allowed triples of values for the latter (000, 111, 001, 100, and 010), so X features 5 branches and none of these branches feature bifurcations.

Finally, the 'global Future' node F just serves to channel out the remaining information. Since it receives all the distinct indices in the graph, its route is just given by the global index constraint. In other words, its practical input set of values is just the set of values permitted by the index-matching and the floating equations.

For the arrows $X \to A$, $X \to B$, and $X \to C$, the 0 value of their index corresponds to a one-dimensional 'dummy' sector. The interpretation of this is once again natural: the messages are sent to an agent only if this agent won. For the arrows $A \to X$, $B \to X$, and $C \to X$, the sectors for which $l = 1$ (resp. $m = 1$, $n = 1$) are trivial as well; this serves to prevent the winner from sending messages to the other agents through the counting station.

We can now check that the routed graph of Figure 27 satisfies our two principles. We start with univocality. The choice relation for this graph can be checked to be a function from the six binary bifurcation choices to the statuses of the branches. This function can be meaningfully presented in the following algorithmic way:

- Look at the votes of the losing branches (A^0, B^0 and C^0). If a majority is found in these votes (say, in favour of Alice), then set the 'result' indices accordingly (in this case, $l = 1$, $m = n = 0$) and use the bifurcation choices of the losing branches (B^0 and C^0) to set the value of the votes of 'losers' (j, k); set the value of the winner's vote to 0 ($i = 0$);

- If no majority is found, define $l = m = n = 0$ and use the bifurcation choices of the

[22] This quirk is necessary to avoid Alice sending information through the vote-counting station if she wins.

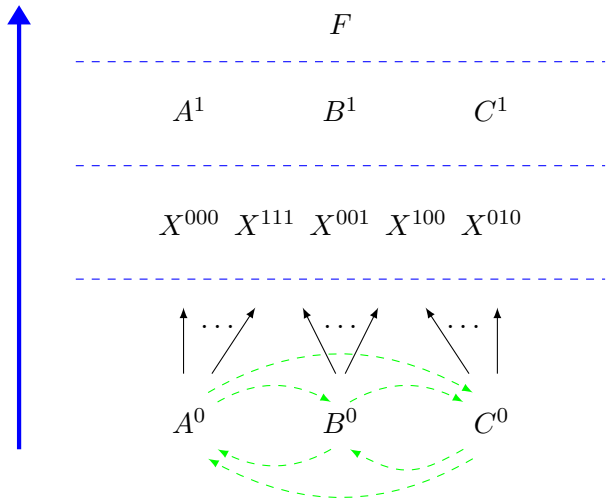


Figure 29: A simplified version of the branch graph for the Lugano process. We do not draw all the arrows, as this would create a lot of clutter and would be superfluous for our purposes of checking for cycles; we rather just organise the branches in layers, such that all unspecified arrows only ever go 'up' with respect to this partition.

losing branches to set all votes.

- Now that the values of all indices in the graph have been fixed, derive which branches happened and which didn't.

Univocality is thus satisfied. Its time-reversed version can be checked to be satisfied as well: all bifurcation choices in the reverse graph are located in F, and they have the effect of fixing all indices to consistent joint values.

A simplified version of the branch graph is presented in Figure 29. We see that there *are* loops in the branch graph, specifically in its bottom layer; yet they are only composed of green dashed arrows. This entails that the routed graph of Figure 27 satisfies the weak loops principle, and is thus valid. In Section 5, we will formulate the conjecture that this presence of weak loops is a signature of its causal inequalities violating nature.

4.3.3 The routed circuit

We proved that our routed graph is a valid one, and therefore that any routed unitary circuit built from it defines a valid process. In particular, the Lugano process (as defined e.g. in equation (27) of Ref. [27]) is obtained by taking all sectors in all wires to be one-dimensional, except for the $l = 1$ sector of the $A \to F$ wire, the $m = 1$ sector of the $B \to F$ wire, and the $n = 1$ sector of the $C \to F$ wire which we take to have dimension 2; and by fleshing out the circuit as depicted in Figure 30. In this figure, V serves to encode a losing agent's vote in the values of the outgoing indices; for example, the V above Alice's node can be written as[23]

$$\begin{aligned} V := &|l=0\rangle\,|i=0,l=0\rangle\,\langle l=0|\,\langle 0| + |l=0\rangle\,|i=1,l=0\rangle\,\langle l=0|\,\langle 1| \\ &+ |l=1,z=0\rangle\,|i=0,l=1\rangle\,\langle l=1|\,\langle 0| + |l=1,z=1\rangle\,|i=0,l=1\rangle\,\langle l=1|\,\langle 1|\ , \end{aligned} \tag{20}$$

W sends the information about the value of its incoming indices to the Future, while also sending the information about the product of those values to the wire that loops back

[23] We denote z for the value of an arbitrary basis of the $l = 1$ two-dimensional sector of $A \to F$.

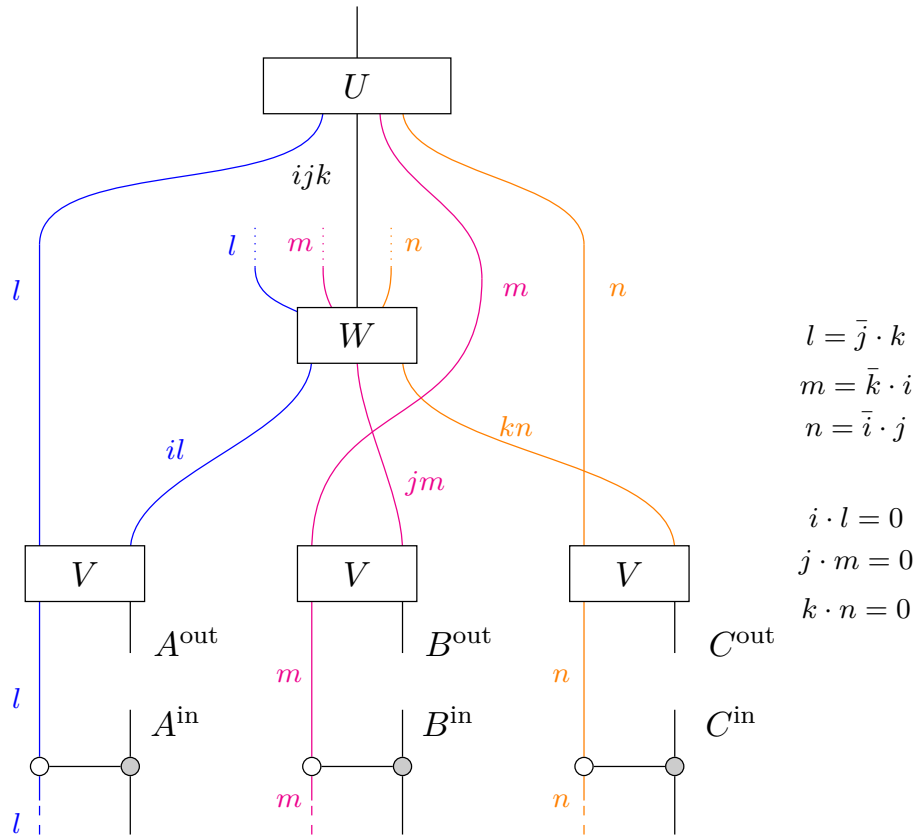


Figure 30: A routed circuit diagram for the Lugano process. We follow the same graphical conventions as in Figure 27. The gates at the bottom are CNOTs, controlled on the coloured wires.

around to the Past:[24]

$$W := \sum_{(i,j,k,l,m,n)\in C} |l\rangle\,|m\rangle\,|ijk\rangle\,|n\rangle\,\langle il|\,\langle jm|\,\langle kn|\ . \tag{21}$$

Finally, U simply embeds its practical input space (defined by the global index constraint) into the global Future. Its precise form is irrelevant to our concerns, so we leave it out.

This shows how a paradigmatic unitary process that violates causal inequalities can be rebuilt using our method. We emphasise, however, that the Lugano process is merely the simplest example of a process obtained from fleshing out a routed circuit of the form of Figure 27; one could instead take this routed circuit to feature arbitrarily large dimensions (as long as the crucial sectors we specified remain one-dimensional), and fill it up with arbitrary operations (as long as they follow the routes). In other words, we have in fact defined a large *family* of processes that all rely on the same core behaviour as the Lugano process.

It is particularly worth noting that, while in the Lugano process the message sent to the winner is trivial (it is necessarily the $|1\rangle$ state), this family of consistent processes includes those where each losing agent can send arbitrarily large messages to the winner. Thus, the routed graph makes clear that the triviality of the messages in the original Lugano process is an arbitrary feature, that is not essential to its consistency.

Readers familiar with [25] might note that the routed circuit we have constructed for

[24] We denote C for the set of tuples (i, j, k, l, m, n) satisfying the graph's global index constraints.

the Lugano process is different from the one in Figure 10 of that work. This comes as no surprise, since the most obvious routed graph from which Figure 10 of [25] can be obtained is not a valid one (c.f. the remark at the end of Section 2.4.2).

5 Discussion and outlook

In this paper, we presented a method for constructing processes with indefinite causal order, based on a decorated directed graph called the routed graph. The method of construction ensures, and hence explains, the consistency of the process, and makes its logical structure evident. In particular, we proved that any process constructed from a routed graph satisfying our two principles, bi-univocality and weak loops, is necessarily consistent.

Our method can be used to construct a number of unitarily extendible processes. We explicitly constructed the quantum switch, the 3-switch, the Grenoble process, and the Lugano process. For each of these processes, our method can also construct a large family of processes obtainable from the same routed graph. We expect that the other currently known examples of unitary processes that are built from classical processes analogous to Lugano can also be constructed using our method. Ultimately, we are led to the following conjecture.

Conjecture 1. *Any unitary process – and therefore any unitarily extendible process – can be obtained by 'fleshing out' a valid routed graph.*

Another fact pointing towards this conjecture is that bipartite unitarily extendible processes were recently proven [25, 28] to reduce to the coherent control of causal orders analogous to the quantum switch, which can therefore be written as valid routed circuits. Our conjecture can be thought of as a tentative generalisation of this result to ≥ 3-partite processes. We expect that significant progress in this direction could be obtained if one were to prove another conjecture, that of the existence of causal decompositions of unitary channels in the general case [18].

This leads us to a limitation of our current results: they offer no systematic way to *decompose* a known process into a routed circuit (except, in some cases, through a careful conceptual analysis of it). An important subject for future work, deeply related to the above conjecture, would be to come up with ways to supplement the bottom-up procedure presented in this paper with a top-down procedure, in which one would start with a 'black-box' unknown process and extract a way of writing it as the fleshing-out of a valid routed graph.

Another limitation is that this paper had no concern for the *physicality* of processes, i.e. for the question of whether and how they could be implemented in practice, using either standard or exotic physics. This was a conscious choice on our part, as we wanted to rather focus on the question of their logical *conceivability*. However, we expect that our way of dealing with the latter question might, through the clarifications and the diagrammatic method it provides, pave the way for work on possible implementations or on physical principles constraining them.

An important consequence of our work is that it shows how at least a large class of valid quantum processes can be derived from the sole study of *possibilistic* structures, encapsulated by routes. These possibilistic structures impose constraints on quantum operations, but there is nothing specifically quantum about them; they could be interpreted as constraints on classical operations as well[25]. This adds to the idea, already conveyed by

[25]Note that *possibilistic* structures also occur in the works of Abramsky and Brandenburger [29], where

the discovery of classical exotic processes, that the logical possibility for indefinite causal order does not always arise from the specifics of quantum structures. If our above conjecture turned out to be true, this would warrant this conclusion for any unitary and unitarily extendible process, whose quantum nature is nothing more than coherence between the branches of an equally admissible classical process.

By contrast, some non-unitarily extendible processes, such as the OCB process [10, 15], appear to feature a more starkly quantum behaviour in their display of indefinite causal order. This can be seen for example in the fact that the violation of causal inequalities by the OCB process relies on a choice between the use of maximally incompatible bases on the part of one agent. A more quantitative clue is the fact that the OCB process saturates a Tsirelson-like bound on non-causal correlations [31, 32]. It is therefore unlikely that such processes could be built using our method, as routes do not capture any specifically quantum (i.e. linear algebraic) behaviour. In particular, the display of a unitary process with OCB-like features would probably provide a counter-example to our conjecture.

In the course of the presentation of the framework and of the main examples, we commented on the fact that the presence of (necessarily weak) loops in the branch graph were associated with the violation of causal inequalities: processes showcasing (possibly dynamical) coherent control of causal order, and therefore incapable of violating causal inequalities [13] – such as the switch, the 3-switch and the Grenoble processes – featured no such loops; while the Lugano process, which does violate causal inequalities, had loops in its branch graph. This leads us to the following conjecture.

Conjecture 2. *The skeletal supermap corresponding to a routed graph violates causal inequalities if and only if its branch graph features (necessarily weak) loops.*

Proving this conjecture would unlock a remarkable correspondence between, on the one hand, the structural features of processes, and, on the other hand, their operational properties. An interesting question is how this would connect to (partial) characterisations of causal inequalities-violating processes via their causal structure [33].

Our work facilitates a transition from a paradigm of defining processes with indefinite causal order one by one and checking their consistency by hand, to one of generating large classes of such processes from the study of elementary graphs, with their consistency baked in. In that, it follows the spirit of Ref. [13], with more emphasis on the connectivity of processes and on the formal language with which one can describe the consistent ones. Another difference is that the framework presented here also allows us to build at least some of the unitary processes that violate causal inequalities [34]. It is important to note that of course, given the matrix form of a purported process in a particular basis, it may or may not be easier to verify if it is a valid process compared to being given the routed graph. However, it is the ability to check the validity for large classes of related processes at once, including processes with arbitrary large dimensions, which lies at the heart of the advantages provided by our method.

A natural application would be to build and study new exotic processes using our framework; we leave this for future work.[26] More generally, the fact that our rules for

they can be viewed as the support of *probabilistic* structures concerning measurement outcomes – see also [30]. In these works, the possibilistic structures pertain to fundamentally quantum phenomena, namely the possibilities for joint measurement outcomes of multipartite quantum states. This is in contrast to the possibilistic structures in our work, which simply denote classical possibilities in the flow of information.

[26]This has in fact been achieved since the first appearance of this paper as a preprint: in [35], a process built using this paper's methods was displayed, featuring a novel unilateral determination of causal order.

validity only rely on the study of graphs decorated with Boolean matrices opens the way for a systematic algorithmic search for instances, using numerical methods.

A final feature of our framework is how, through the use of graphical methods and meaningful principles, it makes more intelligible, and more amenable to intuition, the reasons why a process can be both cyclic and consistent – a notoriously obscure behaviour, especially in the case of processes violating causal inequalities. Our two rules for validity, however, are still high-level; further work is needed to investigate their structural implications. This could eventually lead to a reasoned classification of the graphs that satisfy them, and therefore of (at least a large class of) exotic processes.

Acknowledgments

It is a pleasure to thank Alastair Abbott, Giulio Chiribella, Ognyan Oreshkov, Nicola Pinzani, Eleftherios Tselentis, Julian Wechs, Matt Wilson, and Wataru Yokojima for helpful discussions and comments. AV also thanks Alexandra Elbakyan for her help to access the scientific literature. AV is supported by the EPSRC Centre for Doctoral Training in Controlled Quantum Dynamics. HK acknowledges funding from the UK Engineering and Physical Sciences Research Council (EPSRC) through grant EP/R513295/1. This publication was made possible through the support of the grant 61466 'The Quantum Information Structure of Spacetime (QISS)' (qiss.fr) from the John Templeton Foundation. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the John Templeton Foundation.

References

[1] D. E. Deutsch, "Quantum computational networks," Proceedings of the Royal Society of London: A. Mathematical and Physical Sciences **425** no. 1868, (1989) 73–90.

[2] D. Aharonov, A. Kitaev, and N. Nisan, "Quantum circuits with mixed states," in Proceedings of the thirtieth annual ACM symposium on Theory of computing, pp. 20–30. 1998. `arXiv:quant-ph/9806029`.

[3] S. Abramsky and B. Coecke, "A categorical semantics of quantum protocols," in Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004., pp. 415–425, IEEE. 2004. `arXiv:quant-ph/0402130`.

[4] M. A. Nielsen and I. L. Chuang, Quantum information and quantum computation. Cambridge University Press, 2000.

[5] B. Coecke and A. Kissinger, Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning. Cambridge University Press, 2017.

[6] G. Chiribella, G. M. D'Ariano, and P. Perinotti, "Transforming quantum operations: Quantum supermaps," EPL (Europhysics Letters) **83** no. 3, (2008) 30004, `arXiv:0804.0180 [quant-ph]`.

[7] L. Hardy, "Probability theories with dynamic causal structure: A New framework for quantum gravity," `arXiv:gr-qc/0509120`.

[8] G. Chiribella, G. D'Ariano, P. Perinotti, and B. Valiron, "Beyond quantum computers," `arXiv:0912.0195v1 [quant-ph]`.

[9] G. Chiribella, G. M. D'Ariano, P. Perinotti, and B. Valiron, "Quantum computations without definite causal structure," Physical Review A **88** no. 2, (2013) 022318, arXiv:0912.0195 [quant-ph].

[10] O. Oreshkov, F. Costa, and C. Brukner, "Quantum correlations with no causal order," Nature Communications **3** (2012) 1092, arXiv:1105.4464 [quant-ph].

[11] Ä. Baumeler, A. Feix, and S. Wolf, "Maximal incompatibility of locally classical behavior and global causal order in multiparty scenarios," Physical Review A **90** no. 4, (2014) 042106, arXiv:1403.7333 [quant-ph].

[12] Ä. Baumeler and S. Wolf, "The space of logically consistent classical processes without causal order," New Journal of Physics **18** no. 1, (2016) 013036, arXiv:1507.01714 [quant-ph].

[13] J. Wechs, H. Dourdent, A. A. Abbott, and C. Branciard, "Quantum circuits with classical versus quantum control of causal order," PRX Quantum **2** no. 3, (2021) 030335, arXiv:2101.08796 [quant-ph].

[14] T. Colnaghi, G. M. D'Ariano, S. Facchini, and P. Perinotti, "Quantum computation with programmable connections between gates," Physics Letters A **376** no. 45, (2012) 2940–2943, arXiv:1109.5987 [quant-ph].

[15] M. Araújo, A. Feix, M. Navascués, and Č. Brukner, "A purification postulate for quantum mechanics with indefinite causal order," Quantum **1** (2017) 10, arXiv:1611.08535 [quant-ph].

[16] A. Vanrietvelde, H. Kristjánsson, and J. Barrett, "Routed quantum circuits," Quantum **5** (Jul, 2021) 503, arXiv:2011.08120 [quant-ph].

[17] J. Barrett, R. Lorenz, and O. Oreshkov, "Quantum causal models," arXiv:1906.10726 [quant-ph].

[18] R. Lorenz and J. Barrett, "Causal and compositional structure of unitary transformations," Quantum **5** (2021) 511, arXiv:2001.07774 [quant-ph].

[19] A. Vanrietvelde and G. Chiribella, "Universal control of quantum processes using sector-preserving channels," Quantum Information and Computation **21** no. 15-16, (Dec, 2021) 1320–1352, arXiv:2106.12463 [quant-ph].

[20] M. Wilson and A. Vanrietvelde, "Composable constraints," arXiv:2112.06818 [math.CT].

[21] A. Kissinger and S. Uijlen, "A categorical semantics for causal structure," Logical Methods in Computer Science **Volume 15, Issue 3** (2019) . https://lmcs.episciences.org/5681.

[22] O. Oreshkov, "Time-delocalized quantum subsystems and operations: on the existence of processes with indefinite causal structure in quantum mechanics," Quantum **3** (2019) 206. http://dx.doi.org/10.22331/q-2019-12-02-206.

[23] G. Chiribella and H. Kristjánsson, "Quantum Shannon theory with superpositions of trajectories," Proceedings of the Royal Society A **475** (2019) , arXiv:1812.05292 [quant-ph].

[24] Ämin Baumeler and E. Tselentis, "Equivalence of grandfather and information antinomy under intervention," Electronic Proceedings in Theoretical Computer Science **340** (Sep, 2021) 1–12, arXiv:2004.12921 [quant-ph].

[25] J. Barrett, R. Lorenz, and O. Oreshkov, "Cyclic quantum causal models," Nature Communications **12** no. 1, (2021) 1–15, arXiv:2002.12157 [quant-ph].

[26] A. Joyal, R. Street, and D. Verity, "Traced monoidal categories," Mathematical Proceedings of the Cambridge Philosophical Society **119** no. 3, (1996) 447–468.

[27] M. Araújo, P. A. Guérin, and Ä. Baumeler, "Quantum computation with indefinite causal structures," Physical Review A **96** no. 5, (2017) 052315, arXiv:1706.09854 [quant-ph].

[28] W. Yokojima, M. T. Quintino, A. Soeda, and M. Murao, "Consequences of preserving reversibility in quantum superchannels," Quantum **5** (2021) 441, arXiv:2003.05682 [quant-ph].

[29] S. Abramsky and A. Brandenburger, "The sheaf-theoretic structure of non-locality and contextuality," New Journal of Physics **13** no. 11, (2011) 113036, arXiv:1102.0264 [quant-ph].

[30] T. Fritz, "Possibilistic physics," 2009. fqxi.org/community/forum/topic/569.

[31] C. Brukner, "Bounding quantum correlations with indefinite causal order," New J. Phys. **17** no. 8, (2015) 083034, arXiv:1404.0721 [quant-ph].

[32] Z. Liu and G. Chiribella, "Tsirelson bounds for quantum correlations with indefinite causal order," Nature Communications **16** no. 1, (2025) 3314, arXiv:2403.02749 [quant-ph].

[33] E.-E. Tselentis and Ä. Baumeler, "Admissible Causal Structures and Correlations," PRX Quantum **4** no. 4, (2023) 040307, arXiv:2210.12796 [quant-ph].

[34] C. Branciard, M. Araújo, A. Feix, F. Costa, and Č. Brukner, "The simplest causal inequalities and their violation," New Journal of Physics **18** no. 1, (2015) 013008, arXiv:1508.01704 [quant-ph].

[35] I. Mejdoub and A. Vanrietvelde, "Unilateral determination of causal order in a cyclic process," arXiv:2506.18540 [quant-ph].

[36] A. Jamiołkowski, "Linear transformations which preserve trace and positive semidefiniteness of operators," Reports on Mathematical Physics **3** no. 4, (1972) 275–278.

[37] M.-D. Choi, "Completely positive linear maps on complex matrices," Linear algebra and its applications **10** no. 3, (1975) 285–290.

[38] N. Ormrod, A. Vanrietvelde, and J. Barrett, "Causal structure in the presence of sectorial constraints, with application to the quantum switch," Quantum **7** (2023) 1028, arXiv:2204.10273 [quant-ph].

A The relationship between the supermap and process matrix representations

Two equivalent but distinct mathematical frameworks are in use in the indefinite causal order literature to represent higher-order processes, stemming from two independent lines of work: one is that of supermaps [6, 9] (also called superchannels), and the other is that of process matrices [10, 15] (also called W-matrices). This can lead to some confusion. In this Appendix, we spell out the equivalence between the two pictures, in order to help readers more accustomed to the process matrix picture to translate our results and concepts from the supermap picture, that we use in this paper.

In broad terms, supermaps and process matrices are equivalent mathematical representations of a same higher-order process, connected by the Choi-Jamiołkowski (CJ)

isomorphism. What can add to the confusion is also that they stem from different conceptual points of view on the situations being modelled, and that the equivalence between these points of view might not be obvious at first sight. We will thus start with a conceptual discussion, before spelling out the mathematical equivalence. We will then further comment on how super*unitaries*, which are the focus of this paper, can be translated to super*channels*: the jump is simply the standard one between the linear representation of pure quantum theory and the completely positive representation of mixed quantum theory.

A.1 At the conceptual level

The point of superchannels is to model higher-order transformations, mapping channels to channels, in the same way that channels map states to states. More precisely, in analogy with the fact that channels can be characterised as the only linear mappings $\mathcal{C} \in \mathrm{Lin}\left[\mathrm{Lin}(\mathcal{H}_A^{\mathrm{in}}) \to \mathrm{Lin}(\mathcal{H}_A^{\mathrm{out}})\right]$ that preserve all quantum states – including quantum states on an extended system $\rho \in \mathrm{Lin}[\mathcal{H}_A^{\mathrm{in}} \otimes \mathcal{H}_X]$ –, superchannels are characterised [6] as the linear mappings

$$\mathcal{S} \in \mathrm{Lin}\left[\mathrm{Lin}\left[\mathrm{Lin}(\mathcal{H}_A^{\mathrm{in}}) \to \mathrm{Lin}(\mathcal{H}_A^{\mathrm{out}})\right] \to \mathrm{Lin}\left[\mathrm{Lin}(\mathcal{H}_P) \to \mathrm{Lin}(\mathcal{H}_F)\right]\right] \tag{22}$$

that preserve all quantum channels – including quantum channels on an extended system $\mathcal{C} \in \mathrm{Lin}\left[\mathrm{Lin}(\mathcal{H}_A^{\mathrm{in}} \otimes \mathcal{H}_X) \to \mathrm{Lin}(\mathcal{H}_A^{\mathrm{out}} \otimes \mathcal{H}_Y)\right]$. Moreover, *multipartite* superchannels [8, 9] can act on pairs, or generally tuples, of channels, mapping them to one 'global' channel[27]. The conceptual idea is thus to combine 'Alice's channel' and 'Bob's channel' into a larger channel; it stems from an emphasis on a *computational* picture, focused on the study of architectures for quantum computation.

Another line of research, developed independently, adopts an *operational* picture, insisting instead on the idea of local agents performing quantum measurements – and crucially, getting classical outcomes [10]. Therefore, rather than on a notion of combining operations, it focuses on the task of computing joint probability distributions for these local outcomes. This is where process matrices come in naturally: taking $\mathcal{M}_i \in \mathrm{Lin}[\mathrm{Lin}[\mathcal{H}_A^{\mathrm{in}}] \to \mathrm{Lin}[\mathcal{H}_A^{\mathrm{out}}]]$ as the CP map corresponding to Alice obtaining outcome i, and $M_i \in \mathrm{Lin}[\mathcal{H}_A^{\mathrm{in}} \otimes \mathcal{H}_A^{\mathrm{out}}]$ as its CJ representation (see below for its mathematical definition) – and similarly taking N_j for Bob obtaining the outcome j –, one can write the joint probability compactly as

$$\mathcal{P}(i,j) = \mathrm{Tr}\left[(M_i^T \otimes N_j^T) \circ W\right] , \tag{23}$$

where $W \in \mathrm{Lin}[\mathcal{H}_A^{\mathrm{in}} \otimes \mathcal{H}_A^{\mathrm{out}} \otimes \mathcal{H}_B^{\mathrm{in}} \otimes \mathcal{H}_B^{\mathrm{out}}]$ is the *process matrix*, which one asks to yield well-defined probabilities, through (23), for any choice of measurements on Alice and Bob's parts.

In order to allow for a notion of purification, the process matrix formalism was then extended [15] to model general higher-order operations, with W now also acting on a global past P and a global future F, and the RHS of (23) being taken to be a partial trace on all other systems, so that the LHS yields (the CJ representation of) a quantum operation $P \to F$. One can now see how this gets us closer, at least conceptually, to the notion of a

[27] More precisely, bipartite superchannels were originally defined as acting on the larger space of all non-signalling channels on the tensor product of their two slots. However, it was proven at the same time that the well-defined superchannels on pairs of channels are exactly the same ones as superchannels on non-signalling channels, so we can overlook this difference.

superchannel. The original process matrices as defined in [10] can then be understood as akin to superchannels with a trivial output.

One might, however, worry that this overlooks the key conceptual difference: that process matrices relate not only channels, but also the probabilities of measurement outcomes. Shouldn't that be more general than superchannels? The key idea to understand why this worry is, in fact, unwarranted, is the fact that the obtaining of any measurement outcome can equivalently be modelled as the implementation of a deterministic channel acting on the system at hand and on an ancilla, together with a measurement outcome having been obtained by subsequently measuring the ancilla. In this way, one can recover the probabilities for measurements in the superchannel picture as well.

Before we turn to the mathematical equivalence, let us briefly comment on the respective strengths and weaknesses of the two representations. The main advantage of process matrices is the ability, in an operational picture, to compute joint probabilities in a straightforward and compact way, via the Hilbert-Schmidt inner product of (23). However, this strength becomes a weakness once one is interested in higher-order processes with a non-trivial output: because the process matrix is then a 'flattened' CJ representation of the process – i.e., an operator acting indistinctly on all the input and output Hilbert spaces involved, e.g. on $A^{\text{in}} \otimes A^{\text{out}} \otimes B^{\text{in}} \otimes B^{\text{out}} \otimes P \otimes F$ –, it smears out the distinction i) between the inputs and outputs of local operations (e.g. between A^{in} and A^{out}), and ii) between the inputs and the output of the higher-order transformation (e.g. between A^{in} and A^{out}, which correspond to one of the input channels, and P and F, which correspond to the output channel).

This flattening is the reason why the connectivity of a higher-order process is particularly difficult to parse in a process matrix: identities between systems, for instance, have to be represented not by identity operators but by maximally entangled states. Similarly, the tracing-out move of (23) lends itself badly to intuition and generally messes up the input/output structure. By contrast, the superchannel's type, as a map $\text{Chan}(A^{\text{in}} \to A^{\text{out}}) \times \text{Chan}(B^{\text{in}} \to B^{\text{out}}) \to \text{Chan}(P \to F)$, neatly encodes the different roles of the different spaces, and facilitates intuitions about the connectivity. In particular, as we are especially interested in a faithful and direct representation of the connectivity of processes, we found the superchannel picture more practical for the needs of the present paper.

A.2 At the mathematical level

The process matrix picture relies on the Choi-Jamiolkowski (CJ) representation of CP maps [36, 37], which can be defined as follows. Consider a CP map $\mathcal{M}_A : A^{\text{in}} \to A^{\text{out}}$. We make a copy of the input system A^{in}, and consider the (unnormalised) maximally entangled state $|\Phi^+\rangle = \sum_i |ii\rangle$ on $A^{\text{in}} \otimes A^{\text{in}}$. The CJ representation M_A of $\mathcal{M}_A$ is then the positive operator on $A^{\text{in}} \otimes A^{\text{out}}$ obtained by feeding one half of this entangled state into $\mathcal{M}$:

$$M_A := (\mathcal{I} \otimes \mathcal{M}_A) |\Phi^+\rangle \langle\Phi^+| \tag{24}$$

Process matrices were originally defined as operators mapping CJ representations of CP maps to probabilities via (23). In the bipartite case, they were therefore required to satisfy:

$$\begin{aligned} W \in \text{Lin}[\mathcal{H}_A^{\text{in}} \otimes \mathcal{H}_A^{\text{out}} \otimes \mathcal{H}_B^{\text{in}} \otimes \mathcal{H}_B^{\text{out}}] \\ W \geq 0 \\ \text{Tr}\left[(M^T \otimes N^T) \circ W\right] = 1 \qquad \forall M \forall N \end{aligned} \tag{25}$$

where M and N can be CJ matrices for any pair of *channels* for Alice and Bob. The positivity requirement buys us positive probabilities; the last requirement ensures that our probability distributions are normalised.

In [15], the definition of process matrices was extended so that they output a CJ matrix for a CP map from a 'past' system P to a 'future' system F. In the bipartite case, the extended process matrix $W \in \mathrm{Lin}[\mathcal{H}_P \otimes \mathcal{H}_A^{\mathrm{in}} \otimes \mathcal{H}_A^{\mathrm{out}} \otimes \mathcal{H}_B^{\mathrm{in}} \otimes \mathcal{H}_B^{\mathrm{out}} \otimes \mathcal{H}_F]$ maps CJ matrices M and N to a CJ matrix $G := \mathrm{Tr}_{A^{\mathrm{in}}A^{\mathrm{out}}B^{\mathrm{in}}B^{\mathrm{out}}}\left[(M^T \otimes N^T) \circ W\right] \in \mathrm{Lin}[\mathcal{H}_P \otimes \mathcal{H}_F]$. Now, rather than requiring that we map CP maps to positive and normalised probabilities, we need to require that we map CP maps to CP maps, and channels to channels. This is guaranteed by the following conditions:

$$\begin{aligned} W &\in \mathrm{Lin}[\mathcal{H}_P \otimes \mathcal{H}_A^{\mathrm{in}} \otimes \mathcal{H}_A^{\mathrm{out}} \otimes \mathcal{H}_B^{\mathrm{in}} \otimes \mathcal{H}_B^{\mathrm{out}} \otimes \mathcal{H}_F] \\ W &\geq 0 \\ \mathrm{Tr}_{A^{\mathrm{in}}A^{\mathrm{out}}B^{\mathrm{in}}B^{\mathrm{out}}}\left[(M^T \otimes N^T) \circ W\right] &= G \qquad \forall M \forall N \end{aligned} \tag{26}$$

where M and N represent any channels for Alice and Bob, and we require that G represents a channel from P to F. The definitions for the original and the extended process matrices generalise in an obvious way to the multipartite case.

The original process matrices are special cases of these more general process matrices, in which the global 'past' and 'future' systems P and F are one-dimensional, since in this case probabilities are CP maps and the number 1 is a channel. On the other hand, when one considers any particular state-preparation at P, and traces out F, any of these new, extended process matrices gives rise to a process matrix as originally defined in (25)[13].

We now demonstrate the equivalence of the extended process matrices and superchannels. More precisely, we show that every extended process matrix uniquely defines a valid superchannel, and vice versa. A bipartite process matrix W defines a superchannel $\mathcal{S}$ in the following way:

$$\mathcal{S}(\mathcal{M}, \mathcal{N}) := \mathrm{Choi}^{-1}\left(\mathrm{Tr}_{A^{\mathrm{in}}A^{\mathrm{out}}B^{\mathrm{in}}B^{\mathrm{out}}}\left[(M^T \otimes N^T) \circ W\right]\right) \tag{27}$$

where $\mathcal{M} : \tilde{A}^{\mathrm{in}} \otimes A^{\mathrm{in}} \to \tilde{A}^{\mathrm{out}} \otimes A^{\mathrm{out}}$ is Alice's channel, which also acts on ancillas, and M is its CJ representation, and similarly for $\mathcal{N}$ and N.

One might initially worry that $\mathcal{S}$ need not always be a superchannel, since the extended process matrices were only defined with respect to input CP maps without ancillas, but a superchannel must also preserve channels with ancillas. However, the positivity of W ensures that $G \geq 0$ where

$$\begin{aligned} G :&= \mathrm{Tr}_{A^{\mathrm{in}}A^{\mathrm{out}}B^{\mathrm{in}}B^{\mathrm{out}}}\left[(M^T \otimes N^T) \circ W\right] \\ &\in \mathrm{Lin}[\mathcal{H}_P \otimes \mathcal{H}_{\tilde{A}}^{\mathrm{in}} \otimes \mathcal{H}_{\tilde{B}}^{\mathrm{in}} \otimes \mathcal{H}_F \otimes \mathcal{H}_{\tilde{A}}^{\mathrm{out}} \otimes \mathcal{H}_{\tilde{B}}^{\mathrm{out}}] \end{aligned} \tag{28}$$

A positive CJ matrix always represents a CP map, meaning that $\mathrm{Choi}^{-1}(G)$ is indeed a CP map from the past and ancillary inputs, to the future and ancillary outputs. Then the last condition in (26) ensures that $\mathcal{S}$ maps channels with ancillas to other channels, meaning that $\mathcal{S}$ is indeed a superchannel.

To see how a bipartite superchannel $\mathcal{S}$ defines a process matrix W, we suppose that Alice and Bob both insert swap channels into the superchannel. The process matrix is then the CJ representation of the resulting channel:

$$W := \mathrm{Choi}\left(\mathcal{S}(\mathtt{SWAP}, \mathtt{SWAP})\right) \tag{29}$$

The positivity of W follows from the complete positivity of the channel $\mathcal{S}(\texttt{SWAP}, \texttt{SWAP})$. One can show that the mapping on (CJ representations of) channels provided by W is the same as the mapping provided by $\mathcal{S}$

$$\mathrm{Tr}_{A^{\mathrm{in}}A^{\mathrm{out}}B^{\mathrm{in}}B^{\mathrm{out}}}\left[(M^T \otimes N^T) \circ W\right] = G = \mathrm{Choi}(\mathcal{S}(\mathcal{M}, \mathcal{N})) \tag{30}$$

meaning the last requirement of (26) will also be satisfied.

Rather than superchannels, this work concentrates on what we call *superunitaries* – that is, linear mappings from a set of unitary operators to another unitary operator. To connect these to process matrices, we note that every superunitary uniquely defines a 'unitary superchannel' – that is, a superchannel that always returns a unitary channel when you feed it a set of unitary channels. Given a superunitary S, we define a unitary superchannel in an obvious way:

$$\mathcal{S}(\mathcal{U}, \mathcal{V}) := [S(U, V)]\,(\cdot)\,[S(U, V)]^\dagger \tag{31}$$

That is, the action of $\mathcal{S}$ on the unitary channels $\mathcal{U} := U(\cdot)U^\dagger$ and $\mathcal{V} := V(\cdot)V^\dagger$ (including unitary channels that additionally act on local ancillary systems) is just the channel corresponding to the action of S on U and V. The action of this superchannel on more general channels can then be calculated using the Stinespring dilation. Conversely, any unitary superchannel defines a superunitary, up to an irrelevant global phase.

We have shown that i) superchannels are equivalent to process matrices and ii) superunitaries are equivalent to unitary superchannels (up to phase). Now it follows trivially that superunitaries are equivalent to unitary process matrices – that is, the process matrices that map unitary channels (possibly with ancillas) to unitary channels. As proven in [15], these are precisely the process matrices that can be written in the form $W = |U_W\rangle\langle U_W|$ where

$$|U_W\rangle := (I \otimes U_W)\,|\Phi^+\rangle \tag{32}$$

is a CJ *vector* representing a unitary operator U_W.

B Technical definitions and proofs

B.1 Notations

In this appendix, we spell out our framework in fully technical terms, and prove its central theorem, ensuring that the routed graphs that satisfy the two principles define routed superunitaries.

To do this, and in particular to prove the theorem, we will have to deal with complicated operators, often acting on arbitrary numbers of factored spaces. We therefore introduced some notational techniques to avoid unnecessary clutter, which we will present and motivate here.

The first of these techniques is *padding*. The idea is to have operators act on spaces larger than the ones they were originally defined to act on, simply by tensoring them with identity operators. For instance, we can have $f \in \mathcal{L}(\mathcal{H}_A)$ act on $\mathcal{H}_A \otimes \mathcal{H}_B$, by considering $f \otimes I_B$. However, when – as will often be the case in what follows – there is an arbitrarily large number of factors, and f formally only acts on an arbitrary subset of them, it becomes very heavy notationally, and of limited mathematical interest, to keep track explicitly of which identity operators we should tensor f with. We will thus allow ourselves to make this procedure implicit.

The idea will then be the following: for an operator f, its padded version f_{pad} will be defined as 'f tensored with the identity operators required to make its action meaningful, in the context of the expression at hand'. For instance, taking f to act on $\mathcal{H}_A \otimes \mathcal{H}_B$, and g to act on $\mathcal{H}_B \otimes \mathcal{H}_C$, $g_{\text{pad}} \circ f_{\text{pad}}$ will be taken to mean $(I_A \otimes g) \circ (f \otimes I_C)$, an operator acting on $\mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_C$. For another example, taking h acting on $\mathcal{H}_C \otimes \mathcal{H}_D$, the equation $g_{\text{pad}} \circ f_{\text{pad}} = h_{\text{pad}}$ will be taken to mean $(I_A \otimes g \otimes I_D) \circ (f \otimes I_C \otimes I_D) = I_A \otimes I_B \otimes I_C \otimes h$. This notation will extend to supermaps as well: for instance, if $\mathcal{S}$ is a supermap of type $(\mathcal{H}_A^{\text{in}} \to \mathcal{H}_A^{\text{out}}) \to (\mathcal{H}_P \to \mathcal{H}_F)$, we will define its action on a map $f : \mathcal{H}_A^{\text{in}} \otimes \mathcal{H}_B^{\text{in}} \to \mathcal{H}_A^{\text{out}} \otimes \mathcal{H}_B^{\text{out}}$ as $\mathcal{S}_{\text{pad}}[f]$, which should be understood as $(\mathcal{S} \otimes \mathcal{I})[f]$, where $\mathcal{I}$ is the identity supermap on $\mathcal{L}(\mathcal{H}_B^{\text{in}} \to \mathcal{H}_B^{\text{out}})$.

Another related technique we will use in order to avoid clutter is to disregard the ordering of factors. Indeed, factors in a given tensor product are usually regarded as being labelled by ordered lists, rather than by sets, of indices. For instance, $\mathcal{H}_A \otimes \mathcal{H}_B$ and $\mathcal{H}_B \otimes \mathcal{H}_A$ are usually regarded as different (albeit isomorphic) spaces. Accordingly, suppose we take a map h over $\mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_C$ which decomposes as a tensor product of a map f on $\mathcal{H}_A \otimes \mathcal{H}_C$ and a map g on $\mathcal{H}_B$. This fact, in the usual picture, could not be expressed as $h = f \otimes g$, as the RHS there acts on $\mathcal{H}_A \otimes \mathcal{H}_C \otimes \mathcal{H}_B$. Rather, one should write $h = (I \otimes \texttt{swap}_{C,B}) \circ (f \otimes g) \circ (I \otimes \texttt{swap}_{B,C})$. Another feature of the standard view is that it is not possible to write $\bigotimes_{X \in \{A,B\}} \mathcal{H}_X$, as this expression would leave the order of the factors ambiguous.

For the expressions we will consider, keeping with this use would force us to 1) explicitly introduce arbitrary orderings of all the sets of indices we use to label factors in tensor products, and 2) overload our expressions with swaps, in order to always place next to each other the spaces on which a given map acts. This would once again create a lot of clutter with little relevance. We will therefore abstain ourselves from that constraint, and take the view that tensor products are labelled with sets, rather than ordered lists, of indices. This will allow us to write $h = f \otimes g$ in the case described above, or to write Hilbert spaces of the form $\bigotimes_{X \in \{A,B\}} \mathcal{H}_X$. The expressions we will write in this way could always be recast in the standard view, using arbitrary orderings of the sets at hand and large amounts of swaps.

One might wonder whether either padding or disregarding the ordering of factors might

lead to ambiguities. In fact, such ambiguities only arise if some Hilbert spaces in a tensor product are labelled with the same index, for instance if one is dealing with a Hilbert space like $\mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_A$. This is why we will carefully avoid such situations, by only ever tensoring different – although possibly isomorphic – Hilbert spaces. For instance, if we need a tensor product of $\mathcal{H}_A$ with itself, we will write it as $\mathcal{H}_A \otimes \mathcal{H}_{A'}$, with $\mathcal{H}_A \cong \mathcal{H}_{A'}$.

Note that the same techniques and notations will be used for relations.

B.2 Technical definitions on supermaps

Definition B.1 (Superrelation)**.** *A* <u>*superrelation*</u> *of type* $\times_N (K_N \to L_N) \to (P \to F)$*, where* P*,* F*, the* K_N*'s and the* L_N*'s are all sets, is a relation* $\mathcal{S}^{\mathbf{Rel}} : \mathbf{Rel}(\times_N K_N, \times_{N'} L_{N'}) \to \mathbf{Rel}(P, F)$*. With a slight abuse of notation, we denote* $\mathcal{S}^{\mathbf{Rel}}[(\lambda_N)_N] := \mathcal{S}^{\mathbf{Rel}}[\bigotimes_N \lambda_N]$*.*

Definition B.2 (Supermap)**.** *A* <u>*supermap*</u> *of type* $\times_N (\mathcal{H}_N^{\text{in}} \to \mathcal{H}_N^{\text{out}}) \to (\mathcal{H}_P \to \mathcal{H}_F)$*, where* $\mathcal{H}_P$*,* $\mathcal{H}_F$*, the* $\mathcal{H}_N^{\text{in}}$*'s and the* $\mathcal{H}_N^{\text{out}}$*'s are all finite-dimensional Hilbert spaces, is a linear map* $\mathcal{S} : \mathcal{L}(\bigotimes_N \mathcal{H}_N^{\text{in}}, \bigotimes_{N'} \mathcal{H}_{N'}^{\text{out}}) \to \mathcal{L}(\mathcal{H}_P, \mathcal{H}_F)$*. With a slight abuse of notation, we denote* $\mathcal{S}[(f_N)_N] := \mathcal{S}[\bigotimes_N f_N]$*.*

Definition B.3 (Superunitary)**.** *A* <u>*superunitary*</u> *of type* $\times_N (\mathcal{H}_N^{\text{in}} \to \mathcal{H}_N^{\text{out}}) \to (\mathcal{H}_P \to \mathcal{H}_F)$ *is a supermap of the same type such that, for any choice of ancillary input and output spaces* $\mathcal{H}_N^{\text{in,anc}}$ *and* $\mathcal{H}_N^{\text{out,anc}}$ *at every* N*, and any choice of unitary maps* $U_N : \mathcal{H}_N^{\text{in}} \otimes \mathcal{H}_N^{\text{in,anc}} \to \mathcal{H}_N^{\text{out}} \otimes \mathcal{H}_N^{\text{out,anc}}$ *at every* N*, one has:*

$$\mathcal{S}_{\text{pad}}[(U_N)_N] \text{ is a unitary from } \mathcal{H}_P \otimes \left(\bigotimes_N \mathcal{H}_N^{\text{in,anc}} \right) \text{ to } \mathcal{H}_F \otimes \left(\bigotimes_N \mathcal{H}_N^{\text{out,anc}} \right). \tag{33}$$

Definition B.4 (Routed Supermap)**.** *A* <u>*routed supermap*</u> *of type* $\times_N (\mathcal{H}_N^{\text{in}} \xrightarrow{\lambda_N} \mathcal{H}_N^{\text{out}}) \to (\mathcal{H}_P \xrightarrow{\mu} \mathcal{H}_F)$*, where the* $\mathcal{H}_N^{\text{in}}$*'s and the* $\mathcal{H}_N^{\text{out}}$*'s are sectorised finite-dimensional Hilbert spaces and the* λ_N*'s are relations* $\mathtt{Ind}_N^{\text{in}} \to \mathtt{Ind}_N^{\text{out}}$*, is a supermap: i) which is restricted to only act on the maps of* $\mathcal{L}(\bigotimes_N \mathcal{H}_N^{\text{in}}, \bigotimes_{N'} \mathcal{H}_{N'}^{\text{out}})$ *that follow the route* $\times_N \lambda_N$*; and ii) whose output always follows the route* μ*.*

We say it is superunitary if it satisfies (33) when acting on routed unitaries $U_N : \mathcal{H}_N^{\text{in}} \otimes \mathcal{H}_N^{\text{in,anc}} \xrightarrow{\lambda_N} \mathcal{H}_N^{\text{out}} \otimes \mathcal{H}_N^{\text{out,anc}}$ *that follow the routes.*

B.3 Technical presentation of the framework

Definition B.5 (Indexed graph)**.** *An* <u>*indexed graph*</u> Γ *consists of*

- *a finite set of nodes (or vertices)* $\mathtt{Nodes}_\Gamma$*;*
- *a finite set of arrows (or edges)* $\mathtt{Arr}_\Gamma = \mathtt{Arr}_\Gamma^{\text{in}} \sqcup \mathtt{Arr}_\Gamma^{\text{int}} \sqcup \mathtt{Arr}_\Gamma^{\text{out}}$*;*
- *functions* $\mathtt{head} : \mathtt{Arr}_\Gamma^{\text{in}} \sqcup \mathtt{Arr}_\Gamma^{\text{int}} \to \mathtt{Nodes}_\Gamma$ *and* $\mathtt{tail} : \mathtt{Arr}_\Gamma^{\text{int}} \sqcup \mathtt{Arr}_\Gamma^{\text{out}} \to \mathtt{Nodes}_\Gamma$*;*
- *for each arrow* $A \in \mathtt{Arr}_\Gamma$*, a finite set of indices* $\mathtt{Ind}_A$*, satisfying:* $A \notin \mathtt{Arr}_\Gamma^{\text{int}} \implies \mathtt{Ind}_A$ *is trivial (i.e. is a singleton);*
- *a function* $\mathtt{dim} : \bigsqcup_{A \in \mathtt{Arr}_\Gamma} \mathtt{Ind}_A \to \mathbb{N}^*$*.*

We further define $\mathtt{Ind}_\Gamma^{\mathrm{in}} := \bigtimes_{A\in\mathtt{Arr}_\Gamma^{\mathrm{in}}} \mathtt{Ind}_A$, $\mathtt{Ind}_\Gamma^{\mathrm{out}} := \bigtimes_{A\in\mathtt{Arr}_\Gamma^{\mathrm{out}}} \mathtt{Ind}_A$, and for any $N \in \mathtt{Nodes}_\Gamma$: $\mathrm{in}(N) := \mathtt{head}^{-1}(N)$, $\mathrm{out}(N) := \mathtt{tail}^{-1}(N)$, $\mathtt{Ind}_N^{\mathrm{in}} := \bigtimes_{A\in\mathrm{in}(N)} \mathtt{Ind}_A$ and $\mathtt{Ind}_N^{\mathrm{out}} := \bigtimes_{A\in\mathrm{out}(N)} \mathtt{Ind}_A$.

To prepare for the interpretation of this graph in terms of complex linear maps, we also define the following sectorised Hilbert spaces: for all $A \in \mathtt{Arr}_\Gamma$, $\mathcal{H}_A := \bigoplus_{k\in\mathtt{Ind}_A} \mathcal{H}_A^k$, where $\mathcal{H}_A^k \cong \mathbb{C}^{\mathtt{dim}(k)}$; $\mathcal{H}_P := \bigotimes_{A\in\mathtt{Arr}_\Gamma^{\mathrm{in}}} \mathcal{H}_A = \bigoplus_{\vec{k}\in\mathtt{Ind}_\Gamma^{\mathrm{in}}} \bigotimes \mathcal{H}_A^{k_A}$ and $\mathcal{H}_F := \bigotimes_{A\in\mathtt{Arr}_\Gamma^{\mathrm{out}}} \mathcal{H}_A = \bigoplus_{\vec{k}\in\mathtt{Ind}_\Gamma^{\mathrm{out}}} \bigotimes \mathcal{H}_A^{k_A}$; and for all $N \in \mathtt{Nodes}_\Gamma$, $\mathcal{H}_N^{\mathrm{in}} := \bigotimes_{A\in\mathrm{in}(N)} \mathcal{H}_A = \bigoplus_{\vec{k}\in\mathtt{Ind}_N^{\mathrm{in}}} \bigotimes \mathcal{H}_A^{k_A}$ and $\mathcal{H}_N^{\mathrm{out}} := \bigotimes_{A\in\mathrm{out}(N)} \mathcal{H}_A = \bigoplus_{\vec{k}\in\mathtt{Ind}_N^{\mathrm{out}}} \bigotimes \mathcal{H}_A^{k_A}$.

Definition B.6 (Branched relation)**.** *A relation $\lambda : K \to L$ is said to be <u>branched</u> if, when seen as a function $K \to \mathcal{P}(L)$, it satisfies*

$$\forall k, k' \in K, \lambda(k) \cap \lambda(k') = \lambda(k) \text{ or } \emptyset\,, \tag{34}$$

i.e. $\lambda(k)$ and $\lambda(k')$ are disjoint or the same.

Note that λ is branched if and only if $\lambda^\top$ is branched. Branched relations define compatible, non-complete partitions of their domain and codomain, corresponding to *branches*. Formally, a branch α of the branched relation $\lambda : K \to L$ is a pair of non-empty sets $K^\alpha \subseteq K$ and $L^\alpha \subseteq L$ such that $\lambda(K^\alpha) = L^\alpha$ and $\lambda^\top(L^\alpha) = K^\alpha$. We denote the set of branches of λ as $\mathtt{Bran}(\lambda)$. Note that the partitions are not complete, i.e. $\bigsqcup_{\alpha\in\mathtt{Bran}(\lambda)} K^\alpha$ might not be equal to K (and the same goes for the outputs); the discrepancy corresponds to the indices that are sent by λ to the empty set, as we consider these indices to be part of no branch at all.

Definition B.7 (Routed graph)**.** *A <u>routed graph</u> $(\Gamma, (\lambda_N)_{N\in\mathtt{Nodes}_\Gamma})$ consists of an indexed graph Γ and, for every node N, of a branched relation $\lambda : \mathtt{Ind}_N^{\mathrm{in}} \to \mathtt{Ind}_N^{\mathrm{out}}$, called the route for node N.*

We will write routed graphs as $(\Gamma, (\lambda_N)_N)$ for brevity. We denote elements of $\mathtt{Bran}(\lambda_N)$ as N^α, and denote the set of input (resp. output) indices of N^α as $\mathtt{Ind}_{N^\alpha}^{\mathrm{in}} \subseteq \mathtt{Ind}_N^{\mathrm{in}}$ (resp. $\mathtt{Ind}_{N^\alpha}^{\mathrm{out}} \subseteq \mathtt{Ind}_N^{\mathrm{out}}$). We also define $\mathtt{Bran}_{(\Gamma,(\lambda_N)_N)} := \bigsqcup_{N\in\mathtt{Nodes}_\Gamma} \mathtt{Bran}(\lambda_N)$, the set of all branches in the whole routed graph.

We will now define the notion of a branch being a *strong parent* of another: this will correspond to solid arrows in the branch graph. First, we introduce the set of possible tuple of values of indices, in order to exclude inconsistent assignments of values.

Definition B.8. *We define $\mathtt{PossVal}_\Gamma$ as the subset of $\bigtimes_{A\in\mathtt{Arr}_\Gamma} \mathtt{Ind}_A$ defined by*

$$\forall (k_A)_{A\in\mathtt{Arr}_\Gamma} \in \mathtt{PossVal}_\Gamma, \forall N \in \mathtt{Nodes}_\Gamma,\ (k_A)_{A\in\mathrm{in}(N)} \overset{\lambda_N}{\sim} (k_A)_{A\in\mathrm{out}(N)}\,. \tag{35}$$

A tuple of values is possible if and only if for every node, it yields a input and an output values that are in the same branch.

Lemma B.1. *Let $\vec{k} = (k_A)_{A\in\mathtt{Arr}_\Gamma} \in \bigtimes_{A\in\mathtt{Arr}_\Gamma} \mathtt{Ind}_A$. $\vec{k} \in \mathtt{PossVal}_\Gamma$ if and only if it meets the following two conditions:*

- *$\forall N \in \mathtt{Nodes}_\Gamma$, $(k_A)_{A\in\mathrm{in}(N)}$ is in λ_N's practical inputs and $(k_A)_{A\in\mathrm{out}(N)}$ is in λ_N's practical outputs;*
- *denoting, for every N, $\mu_N^{\mathrm{in}}(\vec{k})$ as the element of $\mathtt{Bran}_N$ such that $(k_A)_{A\in\mathrm{in}(N)} \in \mathtt{Ind}^{\mathrm{in}}_{N^{\mu_N^{\mathrm{in}}(\vec{k})}}$, and $\mu_N^{\mathrm{out}}(\vec{k})$ similarly, we have: $\forall N, \mu_N^{\mathrm{in}}(\vec{k}) = \mu_N^{\mathrm{out}}(\vec{k})$.*

Proof. This derives directly from the structure of the routes: a branched route relates an input value to an output value if and only if they are both in the same branch (and in particular, are not outside of its practical in/outputs). □

For $\vec{k} \in \texttt{PossVal}_\Gamma$, we can therefore denote for every N the branch $\mu_N\left(\vec{k}\right)$ equal to both $\mu_N^{\text{in}}(\vec{k})$ and $\mu_N^{\text{out}}(\vec{k})$.

Definition B.9 (Strong parents)**.** *Let* $(\Gamma, (\lambda_N)_{N\in\texttt{Nodes}_\Gamma})$ *be a routed graph, and* N^α *and* M^β *two branches in it. We define the set of arrows from* N *to* M *as* $\texttt{Link}(N, M) := \text{out}(N) \cap \text{in}(M)$*. We define* $\texttt{LinkVal}(N^\alpha, M^\beta)$*, the set of values linking* N^α *to* M^β*, as*

$$\texttt{LinkVal}(N^\alpha, M^\beta) := \left\{ (k_A)_{A\in\texttt{Link}(N,M)} | \exists (k_A)_{A\in\texttt{Arr}_\Gamma\setminus\texttt{Link}(N,M)} \text{ such that } \begin{cases} \mu_N\left((k_A)_{A\in\texttt{Arr}_\Gamma}\right) = \alpha \\ \mu_M\left((k_A)_{A\in\texttt{Arr}_\Gamma}\right) = \beta \end{cases} \right\} \tag{36}$$

We say that the branch N^α *is* <u>*not a strong parent*</u> *of the branch* M^β *if at least one of the following holds:*

- $\texttt{Link}(N, M) = \emptyset$*;*
- $\texttt{LinkVal}(N^\alpha, M^\beta) = \emptyset$*;*
- $\texttt{LinkVal}(N^\alpha, M^\beta)$ *is a singleton and its unique element* $(k_A)_{A\in\texttt{Link}(N,M)}$ *satisfies* $\forall A \in \texttt{Link}(N, M)$, $\texttt{dim}(k_A) = 1$.

Definition B.10 (Adjoint of a graph)**.** *If* Γ *is an indexed graph, its* <u>*adjoint*</u> $\Gamma^\top$ *is the indexed graph given by swapping the roles of* $\texttt{Arr}_\Gamma^{\text{in}}$ *and* $\texttt{Arr}_\Gamma^{\text{out}}$ *and those of* `head` *and* `tail`*, and leaving the rest invariant.*

If $(\Gamma, (\lambda_N)_N)$ *is a routed graph, its adjoint is the routed graph* $(\Gamma^\top, (\lambda_N{}^\top)_N)$*.*

Definition B.11 (Skeletal superrelation of an indexed graph)**.** *Given an indexed graph* Γ*, its associated* <u>*skeletal superrelation*</u> *is the superrelation* $\mathcal{S}_\Gamma^{\mathbf{Rel}} : \bigtimes_N (\texttt{Ind}_N^{\text{in}} \to \texttt{Ind}_N^{\text{out}}) \to (\texttt{Ind}_\Gamma^{\text{in}} \to \texttt{Ind}_\Gamma^{\text{out}})$ *defined by*

$$\mathcal{S}_\Gamma^{\mathbf{Rel}}[(\lambda_N)_N] := \text{Tr}_{\texttt{Ind}_A, A\in\texttt{Arr}_\Gamma^{\text{int}}}\left[\bigotimes_N \lambda_N\right]. \tag{37}$$

Note that this is well-typed because $\bigtimes_N \texttt{Ind}_N^{\text{in}} = \bigtimes_{A\in\texttt{Arr}_\Gamma^{\text{in}}\sqcup\texttt{Arr}_\Gamma^{\text{int}}} \texttt{Ind}_A$ *and* $\bigtimes_N \texttt{Ind}_N^{\text{out}} = \bigtimes_{A\in\texttt{Arr}_\Gamma^{\text{out}}\sqcup\texttt{Arr}_\Gamma^{\text{int}}} \texttt{Ind}_A$*.*

Definition B.12 (Skeletal supermap of a routed graph)**.** *Given a routed graph* $(\Gamma, (\lambda_N)_N)$*, its associated (routed)* <u>*skeletal supermap*</u> *is the supermap* $\mathcal{S}_{(\Gamma,(\lambda_N)_N)}$ *of type* $\bigtimes_N (\mathcal{H}_N^{\text{in}} \overset{\lambda_N}{\to} \mathcal{H}_N^{\text{out}}) \to (\mathcal{H}_P \overset{\mathcal{S}_\Gamma^{\mathbf{Rel}}[(\lambda_N)_N]}{\to} \mathcal{H}_F)$ *defined by*

$$\mathcal{S}_{(\Gamma,(\lambda_N)_N)}[(f_N)_N] := \text{Tr}_{\mathcal{H}_A, A\in\texttt{Arr}_\Gamma^{\text{int}}}\left[\bigotimes_N f_N\right]. \tag{38}$$

Note that the fact that $\mathcal{S}_{(\Gamma,(\lambda_N)_N)}[(f_N)_N]$ *follows the route* $\mathcal{S}_\Gamma^{\mathbf{Rel}}[(\lambda_N)_N]$ *when the* f_N*'s follow the* λ_N*'s is ensured by the fact that routed maps form a compact closed category [16, 20].*

Definition B.13 (Augmented relation). *Given a relation* $\lambda_N : \mathtt{Ind}_N^{\mathrm{in}} \to \mathtt{Ind}_N^{\mathrm{out}}$ *serving as a route for node* N*, its* <u>*augmented*</u> *version is the partial function (encoded as a relation)* $\lambda_N^{\mathrm{aug}} : \mathtt{Ind}_N^{\mathrm{in}} \times \left(\bigtimes_{\alpha\in\mathtt{Bran}(\lambda_N)} \mathtt{Ind}_{N^\alpha}^{\mathrm{out}}\right) \to \mathtt{Ind}_N^{\mathrm{out}} \times \left(\bigtimes_{\alpha\in\mathtt{Bran}(\lambda_N)} \mathtt{Happens}_{N^\alpha}\right)$ *– where* $\forall\alpha, \mathtt{Happens}_{N^\alpha} \cong \{0,1\}$ *– given by*

$$\lambda_N^{\mathrm{aug}}(k, (l^\alpha)_{\alpha\in\mathtt{Bran}(\lambda_N)}) = \begin{cases} \{(l^\alpha, (\delta^\alpha_{\alpha'})_{\alpha\in\mathtt{Bran}(\lambda_N)})\} & \text{if } k \in \mathtt{Ind}_{N^\alpha}^{\mathrm{in}} \\ \emptyset & \text{if } \forall\alpha, k \notin \mathtt{Ind}_{N^\alpha}^{\mathrm{in}}\,. \end{cases} \tag{39}$$

Definition B.14 (Univocality). *A routed graph* $(\Gamma, (\lambda_N)_N)$ *is* <u>*univocal*</u> *if*

$$\mathcal{S}^{\mathbf{Rel}}_{\Gamma,\mathrm{pad}}\left[(\lambda_N^{\mathrm{aug}})_N\right] \text{ is a function.} \tag{40}$$

We then note this function as $\Lambda_{(\Gamma,(\lambda_N)_N)}$*.*

$(\Gamma, (\lambda_N)_N)$ *is* <u>*bi-univocal*</u> *if both it and its adjoint* $\left(\Gamma^\top, ({\lambda_N}^\top)_N\right)$ *are univocal.*

Definition B.15 (Branch graph). *If* $(\Gamma, (\lambda_N)_N)$ *is a bi-univocal routed graph, its* <u>*branch graph*</u> $\Gamma^{\mathtt{Bran}}$ *is the graph in which*

- *the nodes are the branches of* $(\Gamma, (\lambda_N)_N)$*, i.e. the elements of* $\mathtt{Bran}_{(\Gamma,(\lambda_N)_N)}$*;*
- *there is a green dashed arrow from* N^α *to* M^β *if* $\Lambda_{(\Gamma,(\lambda_N)_N)}$ *features influence from* $\mathtt{Ind}_{N^\alpha}^{\mathrm{out}}$ *to* $\mathtt{Happens}_{M^\beta}$*;*
- *there is a red dashed arrow from* N^α *to* M^β *if* $\Lambda_{(\Gamma^\top,({\lambda_N}^\top)_N)}$ *features influence from* $\mathtt{Ind}_{M^\beta}^{\mathrm{in}}$ *to* $\mathtt{Happens}_{N^\alpha}$*;*
- *there is a solid arrow from* N^α *to* M^β *if* N^α *is a strong parent of* M^β*.*

Definition B.16 (Weak loops). *Let* $(\Gamma, (\lambda_N)_N)$ *be a bi-univocal routed graph. We say that a loop in* $\Gamma^{\mathtt{Bran}}$ *is* <u>*weak*</u> *if it only contains green dashed arrows, or if it only contains red dashed arrows.*

Theorem B.1 (Main theorem). *Let* $(\Gamma, (\lambda_N)_N)$ *be a routed graph which is bi-univocal, and whose branch graph* $\Gamma^{\mathtt{Bran}}$ *only features weak loops. Then its associated skeletal supermap* $\mathcal{S}_{(\Gamma,(\lambda_N)_N)}$ *is a superunitary.*

The rest of this Appendix is dedicated to the proof of this theorem.

B.4 Proof

B.4.1 Preliminary lemmas and definitions

Lemma B.2. *To prove Theorem B.1, it is sufficient to prove that, for any valid routed graph* $(\Gamma, (\lambda_N)_N)$*,* $\mathcal{S}_{(\Gamma,(\lambda_N)_N)}$ *preserves unitarity when acting on input operations without ancillas.*

Proof. Suppose it was proven that for any valid $(\Gamma, (\lambda_N)_N)$, and for any set of routed unitaries $U_N : \mathcal{H}_N^{\mathrm{in}} \overset{\lambda_N}{\to} \mathcal{H}_N^{\mathrm{out}}$, $\mathcal{S}_{(\Gamma,(\lambda_N)_N)}[(U_N)_N]$ is a unitary.

Taking now a valid $(\Gamma, (\lambda_N)_N)$ and, for every N, a choice of ancillary input and output spaces $\mathcal{H}_N^{\mathrm{in,anc}}$ and $\mathcal{H}_N^{\mathrm{out,anc}}$, and a routed unitary $U_N : \mathcal{H}_N^{\mathrm{in}} \otimes \mathcal{H}_N^{\mathrm{in,anc}} \overset{\lambda_N}{\to} \mathcal{H}_N^{\mathrm{out}} \otimes \mathcal{H}_N^{\mathrm{out,anc}}$. One can then define a new indexed graph $\tilde{\Gamma}$ by adding, for each N, a new arrow in $\mathtt{Arr}_\Gamma^{\mathrm{in}}$,

with Hilbert space $\mathcal{H}_N^{\text{in,anc}}$, and a new arrow in $\mathtt{Arr}_\Gamma^{\text{out}}$, with Hilbert space $\mathcal{H}_N^{\text{out,anc}}$. The routed graph $(\tilde{\Gamma}, (\lambda_N)_N)$ then has the same choice relation and the same branch graph as $(\Gamma, (\lambda_N)_N)$; it is therefore valid as well. We can thus apply our assumption to it, which entails that $\mathcal{S}_{(\tilde{\Gamma},(\lambda_N)_N)}[(U_N)_N] = \mathcal{S}^{\text{pad}}_{(\Gamma,(\lambda_N)_N)}[(U_N)_N]$ is unitary. This thus proves the theorem in the general case. □

From now on, we will therefore work with a fixed routed graph $(\Gamma, (\lambda_N)_N)$ (which we will often denote as Γ for simplicity) satisfying bi-univocality and weak loops, and a fixed collection of routed unitary maps $U_N : \mathcal{H}_N^{\text{in}} \overset{\lambda_N}{\to} \mathcal{H}_N^{\text{out}}$ following the λ_N's. Writing $\mathcal{S} := \mathcal{S}_{(\Gamma,(\lambda_N)_N)}$ for simplicity, our goal is to prove that $\mathcal{S}[(U_N)_N] : \mathcal{H}_P \to \mathcal{H}_F$ is a unitary.

For each branch N^α, we define $\mathcal{H}_{N^\alpha}^{\text{in}} := \bigoplus_{(k_A)_{A\in\text{in}(N)}\in\mathtt{Ind}_{N^\alpha}^{\text{in}}} \bigotimes_{A\in\text{in}(N)} \mathcal{H}_A^{k_A} \subseteq \mathcal{H}_N^{\text{in}}$, $\mathcal{H}_{N^\alpha}^{\text{out}} := \bigoplus_{(k_A)_{A\in\text{out}(N)}\in\mathtt{Ind}_{N^\alpha}^{\text{out}}} \bigotimes_{A\in\text{out}(N)} \mathcal{H}_A^{k_A} \subseteq \mathcal{H}_N^{\text{out}}$. We also define the projection $p_N^\alpha : \mathcal{H}_N^{\text{in}} \to \mathcal{H}_{N^\alpha}^{\text{in}}$ and the injection $i_N^\alpha : \mathcal{H}_{N^\alpha}^{\text{out}} \to \mathcal{H}_N^{\text{out}}$.

We define the *exchange* gate for N, $\text{EXCH}_N : \mathcal{H}_N^{\text{in}} \otimes \left(\bigotimes_{\alpha\in\mathtt{Bran}(\lambda_N)} \mathcal{H}_{N^\alpha}^{\text{out}}\right) \to \mathcal{H}^{\text{out}} \otimes \left(\bigotimes_{\alpha\in\mathtt{Bran}(\lambda_N)} \mathcal{H}_{N^\alpha}^{\text{in}}\right)$, by

$$\text{EXCH}_N := \sum_{\alpha\in\mathtt{Bran}(\lambda_N)} i_{N,\text{pad}}^\alpha \circ \left(\text{SWAP}_{N_{\text{in}}^\alpha, N_{\text{out}}^\alpha} \otimes (\bigotimes_{\beta\neq\alpha} \Theta_{N^\beta})\right) \circ p_{N,\text{pad}}^\alpha \,, \tag{41}$$

where $\forall N^\beta$, Θ_{N^β} is an arbitrarily chosen unitary from $\mathcal{H}_{N^\beta}^{\text{out}}$ to $\mathcal{H}_{N^\beta}^{\text{in}}$. Note that EXCH_N follows λ_N by construction.

We note that the fact that the U_N's follow the λ_N's entails that one can find a block decomposition for them, i.e., one can define unitaries $U_N^\alpha : \mathcal{H}_{N^\alpha}^{\text{in}} \to \mathcal{H}_{N^\alpha}^{\text{out}}$ such that

$$\forall N, U_N = \sum_{\alpha\in\mathtt{Bran}(\lambda_N)} i_{N,\text{pad}}^\alpha \circ U_N^\alpha \circ p_{N,\text{pad}}^\alpha \,. \tag{42}$$

As a first preliminary to the proof, we will study in detail how bifurcation choices are in correspondence with assignments of values to the arrow's indices.

The following definition and lemma prove two things. First, univocality implies that any tuple of bifurcation choices fixes not only the branch at every node, but also the specific index values picked in that branch. Second, for a fixed tuple of bifurcation choices, the bifurcation choices at the branches not happening have no effect – i.e. modifying them to any other value wouldn't affect the any of the index values in the graph; while, on the contrary, modifying the bifurcation choice at any of the branches happening always changes at least one of the index values in the graph. In that sense, any tuple of values of the graph's indices corresponds either to no tuple of bifurcation choices at all, or to exactly one bifurcation choice at the branches that happen for this tuple of values, with no dependence on the bifurcation choices at branches that don't happen.

Definition B.17. *For every N in $\mathtt{Nodes}_\Gamma$, we take $\mathtt{Ind}_N^{\text{out}\prime} \cong \mathtt{Ind}_N^{\text{out}}$ and define the partial function (encoded as a relation) $\lambda_N^{\text{sec}} : \mathtt{Ind}_N^{\text{in}} \times \left(\bigtimes_{\alpha\in\mathtt{Bran}_N} \mathtt{Ind}_{N^\alpha}^{\text{out}}\right) \to \mathtt{Ind}_N^{\text{out}} \times \mathtt{Ind}_N^{\text{out}\prime}$ given by*

$$\lambda_N^{\text{sec}}(k, (l^\alpha)_{\alpha\in\mathtt{Bran}_N}) = \begin{cases} \{(l^\alpha, l^\alpha)\} & \text{if } k \in \mathtt{Ind}_{N^\alpha}^{\text{in}} \\ \emptyset & \text{if } \forall\alpha,\, k \notin \mathtt{Ind}_{N^\alpha}^{\text{in}} \,. \end{cases} \tag{43}$$

Lemma B.3. *If $(\Gamma, (\lambda_N)_N)$ is univocal, then $\mathcal{S}^{\mathbf{Rel}}\left[(\lambda_N^{\text{sec}})_N\right]$ is an injective function $\bigtimes_{N^\alpha\in\mathtt{Bran}_\Gamma} \mathtt{Ind}_{N^\alpha}^{\text{out}} \to \bigtimes_N \mathtt{Ind}_N^{\text{out}\prime}$, which we denote Λ^{sec}. Furthermore, its preimage sets are given by*

$$\forall (k_N)_N,\ (\Lambda^{\text{sec}})^{-1}\left((k_N)_N\right) = \text{ either } \emptyset \text{ or } \left(\bigtimes_N \{k_N\}\right) \times \left(\bigtimes_{N^\alpha | k_N \notin \mathtt{Ind}^{\text{out}}_{N^\alpha}} \mathtt{Ind}^{\text{out}}_{N^\alpha}\right). \tag{44}$$

Proof. We will use bra-ket notations for relational states and effects. For every branch N^α, we define $\text{COPY}_{N^\alpha_{\text{out}}} : \mathtt{Ind}^{\text{out}}_{N^\alpha} \to \mathtt{Ind}^{\text{out}}_{N^\alpha} \times \mathtt{Ind}^{\text{out}\prime}_{N^\alpha}$, with $\mathtt{Ind}^{\text{out}\prime}_{N^\alpha} \cong \mathtt{Ind}^{\text{out}}_{N^\alpha}$, by $\text{COPY}_{N^\alpha_{\text{out}}} |l\rangle = |l\rangle \otimes |l\rangle$. For every node N, we define the partial function (encoded as a relation) $\sigma^N : \bigtimes_{\alpha \in \mathtt{Bran}(\lambda_N)} \mathtt{Happens}_{N^\alpha} \to \mathtt{Bran}_N$ by

$$\sigma^N((\varepsilon^{\alpha'})_{\alpha' \in \mathtt{Bran}_N}) = \begin{cases} \{\alpha\} & \text{if } \forall \alpha', \varepsilon^{\alpha'} = \delta^{\alpha'}_\alpha \\ \emptyset & \text{otherwise.} \end{cases} \tag{45}$$

For every node N, we define the function $\nu^N : \mathtt{Bran}_N \times \left(\bigtimes_{\alpha \in \mathtt{Bran}_N} \mathtt{Ind}^{\text{out}\prime}_{N^\alpha}\right) \to \mathtt{Ind}^{\text{out}\prime}_N$, with $\mathtt{Ind}^{\text{out}\prime}_N \cong \mathtt{Ind}^{\text{out}}_N$, by

$$\nu^N(\vec{l}, \alpha) = l^\alpha\,. \tag{46}$$

One can then compute that $\lambda^{\text{sec}}_N = \nu^N_{\text{pad}} \circ \sigma^N_{\text{pad}} \circ \lambda^{\text{aug}}_{N,\text{pad}} \circ \text{COPY}_{N^\alpha_{\text{out}},\text{pad}}$; we can thus reexpress Λ^{sec} in terms of the choice function Λ,

$$\Lambda^{\text{sec}} = \mathcal{S}^{\mathbf{Rel}}\left[(\lambda^{\text{sec}}_N)_N\right] = \left(\prod_N \nu^N_{\text{pad}}\right) \circ \left(\prod_N \sigma^N_{\text{pad}}\right) \circ \Lambda \circ \left(\prod_{N^\alpha} \text{COPY}_{N^\alpha_{\text{out}},\text{pad}}\right). \tag{47}$$

Given that the outputs of a λ^{aug}_N are within the domain of definition of the corresponding σ^N, the fact that Λ is a function thus implies that $\left(\prod_N \sigma^N_{\text{pad}}\right) \circ \Lambda$ is a function as well. Given that the $\text{COPY}_{N^\alpha_{\text{out}}}$'s and ν^N's are functions, Λ^{sec} is a function as well .

Furthermore, let us fix an N and $k_N \in \mathtt{Ind}^{\text{out}}_N$. If k_N is outside of λ_N's practical outputs, it immediately has no preimage through ν^N. Taking the other case, we denote α as the branch of N such that $k_N \in \mathtt{Ind}^{\text{out}}_{N^\alpha}$. Then,

$$\begin{aligned}
&\langle k_N|_{\mathtt{Ind}^{\text{out}\prime}_{N^\alpha},\text{pad}} \circ \lambda^{\text{sec}}_N \\
&= \langle k_N|_{\mathtt{Ind}^{\text{out}\prime}_{N^\alpha},\text{pad}} \circ \nu^N_{\text{pad}} \circ \sigma^N_{\text{pad}} \circ \lambda^{\text{aug}}_{N,\text{pad}} \circ \left(\bigotimes_{\alpha' \in \mathtt{Bran}_N} \text{COPY}_{N^{\alpha'}_{\text{out}}}\right)_{\text{pad}} \\
&= \langle k_N|_{\mathtt{Ind}^{\text{out}\prime}_{N^\alpha},\text{pad}} \circ \left(\bigotimes_{\alpha' \in \mathtt{Bran}_N \setminus \{\alpha\}} \langle \mathtt{Ind}^{\text{out}}_{N^{\alpha'}}|\right)_{\text{pad}} \circ \sigma^N_{\text{pad}} \circ \lambda^{\text{aug}}_{N,\text{pad}} \circ \left(\bigotimes_{\alpha' \in \mathtt{Bran}_N} \text{COPY}_{N^{\alpha'}_{\text{out}}}\right)_{\text{pad}} \\
&= \langle k_N|_{\mathtt{Ind}^{\text{out}\prime}_{N^\alpha},\text{pad}} \circ \sigma^N_{\text{pad}} \circ \lambda^{\text{aug}}_{N,\text{pad}} \circ \left(\,|k_N\rangle\langle k_N|\,\right)_{\mathtt{Ind}^{\text{out}}_{N^\alpha},\text{pad}} \\
&= \left(\bigotimes_{\alpha' \in \mathtt{Bran}_N} \langle \delta^{\alpha'}_\alpha|_{\mathtt{Happens}_{N^{\alpha'}}}\right)_{\text{pad}} \circ \lambda^{\text{aug}}_{N,\text{pad}} \circ \left(\,|k_N\rangle\langle k_N|\,\right)_{\mathtt{Ind}^{\text{out}}_{N^\alpha},\text{pad}} \\
&= |k_N\rangle_{\mathtt{Ind}^{\text{out}}_N} \left(\langle \mathtt{Ind}^{\text{in}}_{N^\alpha}|_{\mathtt{Ind}^{\text{in}}_N} \otimes \langle k_N|_{\mathtt{Ind}^{\text{out}}_{N^\alpha}} \otimes \left(\bigotimes_{\alpha' \in \mathtt{Bran}_N \setminus \{\alpha\}} \langle \mathtt{Ind}^{\text{out}}_{N^{\alpha'}}|\right)\right).
\end{aligned} \tag{48}$$

Therefore, we find that $(\Lambda^{\text{sec}})^{-1}((k_N)_N)$ is empty if at least one of the k_N's is outside of the practical outputs of the corresponding λ_N's, and that otherwise – denoting, for every N, $\alpha(k_N)$ as the branch such that $k_N \in \mathtt{Ind}^{\text{out}}_{N^{\alpha(k_N)}-}$,

$$\begin{aligned}
&\left(\bigotimes_{N\in\mathtt{Nodes}_\Gamma} \langle k_N|_{\mathtt{Ind}^{\text{out}\prime}_N}\right) \circ \Lambda^{\text{sec}} \\
&= \left(\bigotimes_{N\in\mathtt{Nodes}_\Gamma} \langle k_N|_{\mathtt{Ind}^{\text{out}\prime}_N}\right) \circ \mathcal{S}^{\mathbf{Rel}}\left[(\lambda^{\text{sec}}_N)_N\right] \\
&= \mathcal{S}^{\mathbf{Rel}}\left[\left(|k_N\rangle_{\mathtt{Ind}^{\text{out}}_N} \langle \mathtt{Ind}^{\text{in}}_{N^{\alpha(k_N)}}|_{\mathtt{Ind}^{\text{in}}_N}\right)_N\right] \otimes \left(\bigotimes_{N\in\mathtt{Nodes}_\Gamma} \langle k_N|_{\mathtt{Ind}^{\text{out}}_{N^{\alpha(k_N)}}} \otimes \left(\bigotimes_{\alpha'\in\mathtt{Bran}_N\setminus\{\alpha(k_N)\}} \langle \mathtt{Ind}^{\text{out}}_{N^{\alpha'}}|\right)\right)
\end{aligned} \tag{49}$$

$\mathcal{S}^{\mathbf{Rel}}\left[\left(|k_N\rangle_{\mathtt{Ind}^{\text{out}}_N} \langle \mathtt{Ind}^{\text{in}}_{N^{\alpha(k_N)}}|_{\mathtt{Ind}^{\text{in}}_N}\right)_N\right]$ is just a scalar in the theory of relations, i.e. 0 or 1; $(\Lambda^{\text{sec}})^{-1}((k_N)_N)$ is thus non-empty if and only if this scalar is equal to 1, and the rest of the expression yields (44). This also shows that $\Lambda^{\text{sec}}_\Gamma$ is injective.

□

Note that we defined $\Lambda^{\text{sec}}_\Gamma$ as having codomain $\bigtimes_{N\in\mathtt{Nodes}_\Gamma} \mathtt{Ind}^{\text{out}}_N$; but, given that for each N we have $\mathtt{Ind}^{\text{out}}_N = \bigtimes_{A\in\text{out}(N)} \mathtt{Ind}_A$, we can also see it as a function to $\bigtimes_{A\in\mathtt{Arr}_\Gamma} \mathtt{Ind}_A$ (we neglect the discrepancy due to global input arrows of the graph, as their sets of index values are trivial). $\Lambda^{\text{sec}}_\Gamma$ can thus be interpreted as telling us how bifurcation choices fix all indices in the graph. $\Lambda^{\text{sec}}_{\Gamma^\dagger}$, obtained from considering the adjoint graph, tells us the same about reverse bifurcation choices.

From that perspective, in the above Lemma, the case of an empty set of preimages corresponds exactly to impossible assignments of values to the arrows, i.e. to ones that are outside of $\mathtt{PossVal}_\Gamma$.

Lemma B.4. *Given $\vec{k} = (k_A)_{A\in\mathtt{Arr}_\Gamma}$, $(\Lambda^{\text{sec}})^{-1}(\vec{k})$ is empty if and only if $\vec{k} \notin \mathtt{PossVal}_\Gamma$.*

Proof. First, if there exists an N such that $k_N = (k_A)_{A\in\text{out}(N)}$ is outside λ_N's practical outputs, then $(\Lambda^{\text{sec}})^{-1}(\vec{k})$ is empty (as pointed out in the previous proof), and $\vec{k} \notin \mathtt{PossVal}_\Gamma$ (as pointed out in Lemma B.1).

Otherwise, we know from the previous proof that $(\Lambda^{\text{sec}})^{-1}(\vec{k})$ is not empty if and only if $\mathcal{S}^{\mathbf{Rel}}\left[\left(|k_N\rangle_{\mathtt{Ind}^{\text{out}}_N} \langle \mathtt{Ind}^{\text{in}}_{N^{\alpha(k_N)}}|_{\mathtt{Ind}^{\text{in}}_N}\right)_N\right] = 1$. But given how $\mathcal{S}^{\mathbf{Rel}}$ was defined in (37), and the form of the λ_N's, this is the case if and only if for all N, $(k_A)_{A\in\text{in}(N)}$ is in the branch $\alpha(k_N)$. As the function α_N is precisely the function μ^{out}_N defined in Lemma B.1, we thus find the condition $\mu^{\text{out}}_N(\vec{k}) = \mu^{\text{in}}_N(\vec{k})$ showed in this Lemma to be necessary and sufficient for $\vec{k} \in \mathtt{PossVal}_\Gamma$. □

Finally, we draw the consequences of the fact that branches satisfy the weak loops condition. Given a branch N^α, we define the following subsets of $\mathtt{Bran}_\Gamma$. By a 'path' in $\Gamma^{\mathtt{Bran}}$, we mean any sequence of arrows, without a distinction between the solid, green dashed or red dashed types.

- $\mathcal{P}(N^\alpha) := \{O^\gamma \neq N^\alpha \mid \exists \text{ a path } O^\gamma \to N^\alpha \text{ in } \Gamma^{\mathtt{Bran}}\}$, N^α's past;
- $\mathcal{F}(N^\alpha) := \{O^\gamma \neq N^\alpha \mid \exists \text{ a path } N^\alpha \to O^\gamma \text{ in } \Gamma^{\mathtt{Bran}}\}$, N^α's future;

- $\mathcal{L}(N^\alpha) := \mathcal{P}(N^\alpha) \cap \mathcal{F}(N^\alpha)$, N^α's layer (i.e. the branches that form a loop with N^α);
- $\mathcal{P}^{\mathrm{str}}(N^\alpha) := \mathcal{P}(N^\alpha) \setminus \mathcal{L}(N^\alpha)$, N^α's strict past;
- $\mathcal{F}^{\mathrm{str}}(N^\alpha) := \mathcal{F}(N^\alpha) \setminus \mathcal{L}(N^\alpha)$, N^α's strict future;

It is clear that the relation $\sim$, defined by: $N^\alpha \sim O^\gamma$ if $N^\alpha = O^\gamma$ or $O^\gamma \in \mathcal{L}(N^\alpha)$, is an equivalence relation on $\mathtt{Bran}_\Gamma$, partitioning it into a collection of layers. The fact that all loops in $\mathtt{Bran}_\Gamma$ are weak then allows us to say that a given layer either only contains green dashed arrows between its elements (in which case we will call it a green layer), or only contains red dashed arrows (in which case we will call it a red layer)[28].

Furthermore, merging the nodes of each layer transforms $\Gamma^{\mathtt{Bran}}$ into an acyclic graph. One can thus define a partial order between layers. Arbitrarily turning it into a total order, and picking arbitrary orderings within each layer, leads to a total ordering $<$ of $\mathtt{Bran}_\Gamma$ in which branches of a same layer are all next to each other, and in which $N^\alpha < O^\gamma \implies N^\alpha \notin \mathcal{F}^{\mathrm{str}}(O^\gamma)$. We can use this total ordering to label the branches with natural numbers, as $\mathtt{Bran}_\Gamma = \{B(i) \,|\, 1 \leq i \leq n\}$. For a given i and a given branch $N^\alpha > B(i)$, we define $\mathcal{P}_i(N^\alpha) := \mathcal{P}(N^\alpha) \cap \{O^\gamma > B(i)\}$, $\mathcal{F}_i(N^\alpha) := \mathcal{F}(N^\alpha) \cap \{O^\gamma > B(i)\}$, etc.

B.4.2 The induction hypothesis

This ordering of Γ's branches will allows us to define an induction. The point is to start from $\mathcal{S}_{\mathrm{pad}}[(\mathrm{EXCH}_N)_N]$, and then to 'refill' the branches one by one, making sure that the unitary obtained at each step is sufficiently well-behaved to be able to move to the next step. To define it, we will first need to define these 'partially filled exchanges' that are being used at every step i in the induction, which we shall call $V_{N,i}$'s. We do so by defining how they act on each branch: i.e., $\forall i, \forall N^\alpha$, we define $V^\alpha_{N,i} : \mathcal{H}^{\mathrm{in}}_N \otimes \left(\bigotimes_{\beta | N^\beta > B(i)} \mathcal{H}^{\mathrm{out}}_{N^\beta}\right) \to \mathcal{H}^{\mathrm{out}}_N \otimes \left(\bigotimes_{\beta | N^\beta > B(i)} \mathcal{H}^{\mathrm{in}}_{N^\beta}\right)$ by

$$V^\alpha_{N,i} = \begin{cases} i^\alpha_{N,\mathrm{pad}} \circ \left(\mathrm{SWAP}_{N^\alpha_{\mathrm{in}}, N^\alpha_{\mathrm{out}}} \otimes (\bigotimes_{\beta > B(i), \beta \neq \alpha} \Theta_{N^\beta})\right) \circ p^\alpha_{N,\mathrm{pad}} & \text{if } N^\alpha > B(i)\,, \\ (i^\alpha_N \circ U^\alpha_N \circ p^\alpha_N) \otimes \left(\bigotimes_{\beta > B(i)} \Theta_{N^\beta}\right) & \text{if } N^\alpha \leq B(i)\,, \end{cases} \tag{50}$$

and we use them to define

$$V_{N,i} := \sum_{\alpha \in \mathtt{Bran}(\lambda_N)} V^\alpha_{N,i}\,. \tag{51}$$

We will write the input (resp. output) space of $\mathcal{S}_{\mathrm{pad}}[(V_{N,i})_N]$ as $\mathcal{H}^{\mathrm{out}}_i := \mathcal{H}_P \otimes \left(\bigotimes_{N^\beta > B(i)} \mathcal{H}^{\mathrm{out}}_{N^\beta}\right)$ (resp. $\mathcal{H}^{\mathrm{in}}_i := \mathcal{H}_F \otimes \left(\bigotimes_{N^\beta > B(i)} \mathcal{H}^{\mathrm{in}}_{N^\beta}\right)$). We also write $\bar{V}^\alpha_{N,i} := V_{N,i} - V^\alpha_{N,i}$. Note that the $V_{N,i}$'s follow the λ_N's by construction, and that one has $V_{N,0} = \mathrm{EXCH}_N$ and $V_{N,n} = U_N$.

The core of the induction will be the hypothesis that, at step i, $\mathcal{S}_{\mathrm{pad}}[(V_{N,i})_N]$ is unitary. However, this will not be sufficient: we will also need other conditions ensuring that $\mathcal{S}_{\mathrm{pad}}[(V_{N,i})_N]$ features structural properties which allow us to move to step $i+1$. More precisely, these conditions will encode the fact that at every step i, and for every branch N^α that hasn't been filled yet (i.e. such that $N^\alpha > B(i)$), one can find projectors on

[28] Note that single-branch layers can be considered to be either green or red: the choice will not affect the proof.

$\mathcal{S}_{\text{pad}}[(V_{N,i})_N]$'s inputs and outputs that control whether N^α happens or not, and that all these projectors will play well with one another.

One subtlety is that, if $B(i)$ is in a *red* layer and if there are still unfilled branches in that layer, then the projectors controlling the status of branches above that layer cannot be defined. This is ultimately not problematic, as one can wait for the whole layer to have been filled to redefine them; but this will force us to amend parts of the induction hypothesis when it is the case.

Finally, another part of the induction hypothesis will rely on the causal properties of $\mathcal{S}_{\text{pad}}\left[(V_{N,i})_N\right]$. We will describe these by using the behaviour of $\mathcal{S}_{\text{pad}}[(V_{N,i})_N]$ seen as an isomorphism of operator algebras, defining $\mathcal{V}_i : \text{Lin}\left[\mathcal{H}_P \otimes \left(\bigotimes_{O^\gamma > B(i)} \mathcal{H}_{O^\gamma_{\text{out}}}\right)\right] \to \text{Lin}\left[\mathcal{H}_F \otimes \left(\bigotimes_{O^\gamma > B(i)} \mathcal{H}_{O^\gamma_{\text{in}}}\right)\right]$ by

$$\forall f, \mathcal{V}_i[f] := \mathcal{S}_{\text{pad}}[(V_{N,i})_N] \circ f \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]^\dagger \,. \tag{52}$$

When $\mathcal{S}_{\text{pad}}[(V_{N,i})_N]$ is unitary, this defines an isomorphism of operator algebras, preserving sums, compositions, and the dagger. This implies that, more generally, $\mathcal{V}_i$ will preserve commutation relations, self-adjointness, idempotency, etc.

We now turn to our induction hypotheses at step i.

Induction Hypothesis 1 (H1)**.** $\mathcal{S}_{\text{pad}}[(V_{N,i})_N]$ *is unitary.*

As we mentioned, H1 is the core of the induction, and will allow us to conclude in the end that $\mathcal{S}[(V_{N,n})_N] = \mathcal{S}[(U_N)_N]$ is indeed unitary.

Induction Hypothesis 2 (H2)**.** *One has defined, for all $N^\alpha > B(i)$, orthogonal projectors:*

- $\zeta_i^{\text{out}}(N^\alpha)$*, acting on* $\begin{cases} \mathcal{H}_P \otimes \left(\bigotimes_{O^\gamma \in \mathcal{P}_i(N^\alpha)} \mathcal{H}_{O^\gamma_{\text{out}}}\right) & \text{if } N^\alpha \text{ is in a green layer;} \\ \mathcal{H}_P \otimes \left(\bigotimes_{O^\gamma \in \mathcal{P}_i^{\text{str}}(N^\alpha)} \mathcal{H}_{O^\gamma_{\text{out}}}\right) & \text{if } N^\alpha \text{ is in a red layer;} \end{cases}$
- $\zeta_i^{\text{in}}(N^\alpha)$*, acting on* $\begin{cases} \mathcal{H}_F \otimes \left(\bigotimes_{O^\gamma \in \mathcal{F}_i^{\text{str}}(N^\alpha)} \mathcal{H}_{O^\gamma_{\text{in}}}\right) & \text{if } N^\alpha \text{ is in a green layer;} \\ \mathcal{H}_F \otimes \left(\bigotimes_{O^\gamma \in \mathcal{F}_i(N^\alpha)} \mathcal{H}_{O^\gamma_{\text{in}}}\right) & \text{if } N^\alpha \text{ is in a red layer;} \end{cases}$

such that (once correctly padded) the $\zeta_{i,\text{pad}}^{\text{out}}(N^\alpha)$'s for different N^α's all commute pairwise, and the $\zeta_{i,\text{pad}}^{\text{in}}(N^\alpha)$'s commute as well, and such that

$$\forall N^\alpha,\, \zeta_{i,\text{pad}}^{\text{in}}(N^\alpha) = \mathcal{V}_i[\zeta_{i,\text{pad}}^{\text{out}}(N^\alpha)] \,. \tag{53}$$

If $B(i)$ and $B(i+1)$ are in the same red layer, then all of the former definitions have only been made for the N^α's in that same layer, i.e. in $\mathcal{L}_i(B(i))$. When this happens, we say that <u>i is a special step</u>.

H2 introduces the projectors that will be used to control the status of the still-unfilled branches. The fact that the ζ's commute pairwise ensures that these controls can always be meaningfully combined. Note that the out-projector for N^α only acts on N^α's past, while its in-projector only acts on N^α's future; and furthermore, that for N^α in a green layer its in-projector only acts on N^α's *strict* future, while for N^α in a red layer its out-projector only acts on its *strict* past. In particular, the $\zeta(N^\alpha)$'s never act on N^α itself: this ensures that at any step, a branch never holds some part of its own controls.

We will also write $\bar{\zeta}_i^{\text{in}}(N^\alpha) := \mathbb{1} - \zeta_i^{\text{in}}(N^\alpha)$ and $\bar{\zeta}_i^{\text{out}}(N^\alpha) := \mathbb{1} - \zeta_i^{\text{out}}(N^\alpha)$.

Induction Hypothesis 3 (H3)**.** *The ζ_i^{out}'s satisfy*

$$\forall N^\alpha, \forall O^\gamma, \zeta_{i,\text{pad}}^{\text{out}}(N^\alpha) \circ \bar{\zeta}_{i,\text{pad}}^{\text{out}}(O^\gamma) \text{ acts trivially on } \mathcal{H}_{O^\gamma_{\text{out}}}\,, \tag{54}$$

and the ζ_i^{in}'s satisfy

$$\forall N^\alpha, \forall O^\gamma, \zeta_{i,\text{pad}}^{\text{in}}(N^\alpha) \circ \bar{\zeta}_{i,\text{pad}}^{\text{in}}(O^\gamma) \text{ acts trivially on } \mathcal{H}_{O^\gamma_{\text{in}}}\,. \tag{55}$$

This hypothesis encodes the fact that, when a branch O^γ doesn't happen, it doesn't hold any control on other branches N^α. Note that the $\forall N^\alpha, O^\gamma$ only runs over the branches for which the ζ's have been defined in H2, i.e. it only runs over $\mathcal{L}_i(B(i))$ if i is a special step. The same will apply in the other hypotheses.

Induction Hypothesis 4 (H4)**.** *The ζ_i^{out}'s satisfy:*

$$\forall N^\alpha, N^\beta, \text{ branches of the same node, } \zeta_{i,\text{pad}}^{\text{out}}(N^\alpha) \circ \zeta_{i,\text{pad}}^{\text{out}}(N^\beta) = 0\,. \tag{56}$$

The meaning is that two branches of the same node are incompatible. Note that one can infer, using (53), that the ζ_i^{in}'s then satisfy the same property.

Induction Hypothesis 5 (H5)**.** *Let $Q \subseteq \{B(i') \,|\, i' \geq i\}$ a set of branches on different nodes; i.e., one can define $\tilde{Q} \subseteq \texttt{Nodes}_\Gamma$ and a function α such that $Q = \{N^{\alpha(N)} \,|\, N \in \tilde{Q}\}$. Then,*

$$\mathcal{S}_{\text{pad}}[(V_{N,i})_N] \circ \prod_{N\in\tilde{Q}} \zeta_{i,\text{pad}}^{\text{out}}\left(N^{\alpha(N)}\right) = \mathcal{S}_{\text{pad}}\left[(V_{N,i})_{N\in\texttt{Nodes}_\Gamma\setminus\tilde{Q}} \times \left(V_{N,i}^{\alpha(N)}\right)_{N\in\tilde{Q}}\right]. \tag{57}$$

H5 formalises the fact that the ζ_i^{out}'s control whether branches happen or not. Note that, using (53), one could have written the same equation using ζ_i^{in}'s.

Induction Hypothesis 6 (H6)**.** *For a branch N^α in a green layer, we have*

$$\begin{aligned}&\forall f \in \text{Lin}[\mathcal{H}_{N^\alpha_{\text{in}}}], \exists f' \in \text{Lin}\left[\mathcal{H}_P \otimes \left(\bigotimes_{O^\gamma\in\mathcal{P}_i^{\text{str}}(N^\alpha)} \mathcal{H}_{O^\gamma_{\text{out}}}\right)\right] \text{ such that}\\ &\mathcal{V}_i^\dagger[f_{\text{pad}}] \circ \zeta_{i,\text{pad}}^{\text{out}}(N^\alpha) = {f'}_{\text{pad}} \circ \zeta_{i,\text{pad}}^{\text{out}}(N^\alpha)\,.\end{aligned} \tag{58}$$

H6 means that, for a branch N^α in a green layer, provided that one is in the subspace in which branch N^α happens, $\mathcal{S}_{\text{pad}}[(V_{N,i})_N]$'s causal structure only has the *strict* past of N^α signalling to N^α_{in}. This will be important to ensure that, when N^α is 'refilled', the action of any ζ_i^{out}'s on it becomes an action on its strict past.

Induction Hypothesis 7 (H7)**.** *For a branch N^α in a red layer, we have*

$$\begin{aligned}&\forall f \in \text{Lin}[\mathcal{H}_{N^\alpha_{\text{out}}}], \exists f' \in \text{Lin}\left[\mathcal{H}_F \otimes \left(\bigotimes_{O^\gamma\in\mathcal{F}_i^{\text{str}}(N^\alpha)} \mathcal{H}_{O^\gamma_{\text{in}}}\right)\right] \text{ such that}\\ &\mathcal{V}_i[f_{\text{pad}}] \circ \zeta_{i,\text{pad}}^{\text{in}}(N^\alpha) = {f'}_{\text{pad}} \circ \zeta_{i,\text{pad}}^{\text{in}}(N^\alpha)\,.\end{aligned} \tag{59}$$

H7 plays the same role as H6 in the reverse time direction.

B.4.3 Proof of the base case

H1 The proof that $\mathcal{S}_{\text{pad}}[(\text{EXCH}_N)_N]$ is unitary will rely on the lemmas of Section B.4.1. To use them, we will first introduce a way to show how bifurcation choices are enforced through the use of the EXCH's. For every A in $\mathtt{Arr}_\Gamma$, we define $\text{WITNESS}_A : \mathcal{H}_A \to \mathcal{H}_A \otimes \mathbb{C}^{|\mathtt{Ind}_A|}$ by

$$\text{WITNESS}_A := \sum_{k_A \in \mathtt{Ind}_A} \pi_A^{k_A} \otimes |k_A\rangle \ , \tag{60}$$

where the $\pi_A^{k_A}$'s are the projectors on the different sectors of A, and we've introduced an arbitrary basis of $\mathbb{C}^{|\mathtt{Ind}_A|}$ labelled by A's index values. The point of WITNESS_A is simply to channel out the information about each arrow's index value.

For a given N, with respect to the sectorisations of the $\mathcal{H}_{N^\alpha_{\text{out}}}$, of the $\mathcal{H}_A$'s for the A's in $\text{in}(N)$ and $\text{out}(N)$, and to the sectorisation of the $\mathbb{C}^{|\mathtt{Ind}_A|}$'s given by the previous basis, λ_N^{sec} is a route for $\left(\bigotimes_{A\in\text{out}(N)} \text{WITNESS}_A\right)_{\text{pad}} \circ \text{EXCH}_N$. Thus (because the compatibility with routes is preserved by the dagger compact structure [16]), $\mathcal{S}^{\mathbf{Rel}}_{\text{pad}}[(\lambda_N^{\text{sec}})_N] = \Lambda_\Gamma^{\text{sec}}$ is a route for $\mathcal{S}_{\text{pad}}\left[\left(\left(\bigotimes_{A\in\text{out}(N)} \text{WITNESS}_A\right)_{\text{pad}} \circ \text{EXCH}_N\right)_N\right]$. Therefore,

$$\begin{aligned}
&\mathcal{S}_{\text{pad}}\left[\left(\left(\bigotimes_{A\in\text{out}(N)} \text{WITNESS}_A\right)_{\text{pad}} \circ \text{EXCH}_N\right)_N\right] \\
&= \sum_{\vec{k}\in\times_A \mathtt{Ind}_A} \left(\bigotimes_A |k_A\rangle\langle k_A|\right)_{\text{pad}} \circ \mathcal{S}_{\text{pad}}\left[\left(\left(\bigotimes_{A\in\text{out}(N)} \text{WITNESS}_A\right)_{\text{pad}} \circ \text{EXCH}_N\right)_N\right] \\
&\circ \left(\sum_{\vec{q}\in(\Lambda^{\text{sec}})^{-1}(\vec{k})} \bigotimes_{N^\alpha\in\mathtt{Bran}_\Gamma} \pi^{q_{N^\alpha}}_{N^\alpha_{\text{out}}}\right)_{\text{pad}} \\
&\overset{\text{Lemma B.4}}{=} \sum_{\vec{k}\in\mathtt{PossVal}_\Gamma} \left(\bigotimes_A |k_A\rangle\langle k_A|\right)_{\text{pad}} \circ \mathcal{S}_{\text{pad}}\left[\left(\left(\bigotimes_{A\in\text{out}(N)} \text{WITNESS}_A\right)_{\text{pad}} \circ \text{EXCH}_N\right)_N\right] \\
&\circ \left(\sum_{\vec{q}\in(\Lambda^{\text{sec}})^{-1}(\vec{k})} \bigotimes_{N^\alpha\in\mathtt{Bran}_\Gamma} \pi^{q_{N^\alpha}}_{N^\alpha_{\text{out}}}\right)_{\text{pad}} \\
&\overset{\text{Lemma B.3}}{=} \sum_{\vec{k}\in\mathtt{PossVal}_\Gamma} \left(\bigotimes_A |k_A\rangle\langle k_A|\right)_{\text{pad}} \circ \mathcal{S}_{\text{pad}}\left[\left(\left(\bigotimes_{A\in\text{out}(N)} \text{WITNESS}_A\right)_{\text{pad}} \circ \text{EXCH}_N\right)_N\right] \\
&\circ \left(\bigotimes_{N\in\mathtt{Nodes}_\Gamma} \pi^{(k_A)_{A\in\text{out}(N)}}_{N^{\mu_N}_{\text{out}}(\vec{k})}\right)_{\text{pad}} ,
\end{aligned} \tag{61}$$

where $\pi^{(k_A)_{A\in\text{out}(N)}}_{N^{\mu_N}_{\text{out}}(\vec{k})}$ is the projector on $\mathcal{H}_{N^{\mu_N}_{\text{out}}(\vec{k})}$'s sector labelled by $(k_A)_{A\in\text{out}(N)}$ (remember that for a given N^α, we have $\mathcal{H}_{N^\alpha_{\text{out}}} = \bigoplus_{(k_A)_{A\in\text{out}(N)}\in\mathtt{Ind}^{\text{out}}_{N^\alpha}} \bigotimes_{A\in\text{out}(N)} \mathcal{H}_A^{k_A}$).

Moreover, we have $\left(\sum_{k\in\mathtt{Ind}_A} \langle k|\right)_{\text{pad}} \circ \text{WITNESS}_A = \mathbb{1}_A$, so

$$
\begin{aligned}
&\mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_N\right] \\
&= \mathcal{S}_{\text{pad}}\left[\left(\sum_{(k_A)_A\in\mathtt{Ind}_N^{\text{out}}}\left(\bigotimes_{A\in\text{out}(N)}\langle k_A|\circ\text{WITNESS}_A\right)_{\text{pad}}\text{EXCH}_N\right)_N\right] \\
&= \sum_{\vec{k}\in\times_A\mathtt{Ind}_A}\left(\bigotimes_A\langle k_A|\right)_{\text{pad}}\mathcal{S}_{\text{pad}}\left[\left(\left(\text{WITNESS}_A\right)_{\text{pad}}\circ\text{EXCH}_N\right)_N\right] \\
&\overset{(61)}{=} \sum_{\vec{k}\in\mathtt{PossVal}_\Gamma}\left(\bigotimes_A\langle k_A|\right)_{\text{pad}}\circ\mathcal{S}_{\text{pad}}\left[\left(\left(\bigotimes_{A\in\text{out}(N)}\text{WITNESS}_A\right)_{\text{pad}}\circ\text{EXCH}_N\right)_N\right] \\
&\circ\left(\bigotimes_{N\in\mathtt{Nodes}_\Gamma}\pi^{(k_A)_{A\in\text{out}(N)}}_{N_{\text{out}}^{\mu_N(\vec{k})}}\right)_{\text{pad}} \\
&= \sum_{\vec{k}\in\mathtt{PossVal}_\Gamma}\mathcal{S}_{\text{pad}}\left[\left(\left(\bigotimes_{A\in\text{out}(N)}\pi_A^{k_A}\right)_{\text{pad}}\circ\text{EXCH}_N\right)_N\right]\circ\left(\bigotimes_{N\in\mathtt{Nodes}_\Gamma}\pi^{(k_A)_{A\in\text{out}(N)}}_{N_{\text{out}}^{\mu_N(\vec{k})}}\right)_{\text{pad}}.
\end{aligned}
\tag{62}
$$

A symmetric argument relying on $\Gamma^\top$ leads to

$$
\begin{aligned}
&\mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_N\right] \\
&= \sum_{\vec{k}\in\mathtt{PossVal}_\Gamma}\left(\bigotimes_{N\in\mathtt{Nodes}_\Gamma}\pi^{(k_A)_{A\in\text{in}(N)}}_{N_{\text{in}}^{\mu_N(\vec{k})}}\right)_{\text{pad}}\circ\mathcal{S}_{\text{pad}}\left[\left(\text{EXCH}_N\circ\left(\otimes_{A\in\text{in}(N)}\pi_A^{k_A}\right)_{\text{pad}}\right)_N\right].
\end{aligned}
\tag{63}
$$

Furthermore, the projectors $\left(\bigotimes_{N\in\mathtt{Nodes}_\Gamma}\pi^{(k_A)_{A\in\text{out}(N)}}_{N_{\text{out}}^{\mu_N(\vec{k})}}\right)_{\text{pad}}$, for $\vec{k}\in\mathtt{PossVal}_\Gamma$, form a sectorisation of the input space of $\mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_N\right]$. Indeed, by Lemma B.3, the

$$
\left(\bigtimes_N\{(k_A)_{A\in\text{out}(N)}\}_{\mathtt{Ind}^{\text{out}}_{N^{\mu_N(\vec{k})}}}\right)\times\left(\bigtimes_{N^\alpha|\alpha\neq\mu_N(\vec{k})}\mathtt{Ind}^{\text{out}}_{N^\alpha}\right)
\tag{64}
$$

are the preimage sets of the injective function $\Lambda_\Gamma^{\text{sec}}$, and therefore form a partition of its domain $\bigtimes_{N^\alpha}\mathtt{Ind}^{\text{out}}_{N^\alpha}$. The sectorisation is thus obtained as a coarse-graining of that given by the $\bigotimes_{N^\alpha}\pi^{(k_A)_{A\in\text{out}(N)}}_{N^\alpha_{\text{out}}}$. Symmetrically, the $\left(\bigotimes_{N\in\mathtt{Nodes}_\Gamma}\pi^{(k_A)_{A\in\text{in}(N)}}_{N_{\text{in}}^{\mu_N(\vec{k})}}\right)_{\text{pad}}$ form a sectorisation of $\mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_N\right]$'s codomain. Crucially, $\mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_N\right]$ is block diagonal with respect to these two sectorisations: indeed, for a given $\vec{k}$,

$$
\begin{aligned}
&\mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_N\right] \circ \left(\bigotimes_{N\in\texttt{Nodes}_\Gamma} \pi^{(k_A)_{A\in\text{out}(N)}}_{N^{\mu_N(\vec{k})}_{\text{out}}}\right)_{\text{pad}} \\
&\overset{(62)}{=} \mathcal{S}_{\text{pad}}\left[\left(\left(\bigotimes_{A\in\text{out}(N)} \pi^{k_A}_A\right)_{\text{pad}} \circ \text{EXCH}_N\right)_N\right] \\
&\overset{(38)}{=} \mathcal{S}_{\text{pad}}\left[\left(\text{EXCH}_N \circ \left(\bigotimes_{A\in\text{in}(N)} \pi^{k_A}_A\right)_{\text{pad}}\right)_N\right] \\
&\overset{(63)}{=} \left(\bigotimes_{N\in\texttt{Nodes}_\Gamma} \pi^{(k_A)_{A\in\text{in}(N)}}_{N^{\mu_N(\vec{k})}_{\text{in}}}\right)_{\text{pad}} \circ \mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_N\right]
\end{aligned}
\tag{65}
$$

All that is left for us to prove is that all of these blocks, which we will denote as $T^{\vec{k}}$'s, are unitary (with respect to the suitable restrictions of their domain and codomain). We start by computing

$$
\begin{aligned}
T^{\vec{k}} &= \mathcal{S}_{\text{pad}}\left[\left(\left(\bigotimes_{A\in\text{out}(N)} \pi^{k_A}_A\right)_{\text{pad}} \circ \text{EXCH}_N\right)_N\right] \\
&\overset{(41)}{=} \mathcal{S}_{\text{pad}}\left[\left(\left(\bigotimes_{A\in\text{out}(N)} \pi^{k_A}_A\right)_{\text{pad}} \circ i^{\mu_N(\vec{k})}_{N,\text{pad}} \circ \left(\text{SWAP}_{N^{\mu_N(\vec{k})}_{\text{in}},N^{\mu_N(\vec{k})}_{\text{out}}} \otimes \left(\bigotimes_{\beta\neq\mu_N(\vec{k})} \Theta_{N^\beta}\right)\right) \circ p^{\mu_N(\vec{k})}_{N,\text{pad}}\right)_N\right] \\
&\overset{(38)}{=} \text{Tr}_{A\in\texttt{Arr}^{\text{int}}_\Gamma}\left[\bigotimes_N \left(\bigotimes_{A\in\text{out}(N)} \pi^{k_A}_A\right)_{\text{pad}} \circ i^{\mu_N(\vec{k})}_{N,\text{pad}} \circ \text{SWAP}_{N^{\mu_N(\vec{k})}_{\text{in}},N^{\mu_N(\vec{k})}_{\text{out}}} \circ p^{\mu_N(\vec{k})}_{N,\text{pad}}\right] \otimes \left(\bigotimes_{M,\beta\neq\mu_M(\vec{k})} \Theta_{M^\beta}\right) \\
&= \left(\left(\bigotimes_N p^{\mu_N(\vec{k})}_N\right) \circ \left(\bigotimes_{A\in\texttt{Arr}_\Gamma} \pi^{k_A}_A\right) \circ \left(\bigotimes_N i^{\mu_N(\vec{k})}_N\right)\right) \otimes \left(\bigotimes_{M,\beta\neq\mu_M(\vec{k})} \Theta_{M^\beta}\right) .
\end{aligned}
\tag{66}
$$

Remember that $i^{\mu_N(\vec{k})}_N$ is the injection $\mathcal{H}^{\text{out}}_{N^{\mu_N(\vec{k})}} \to \mathcal{H}^{\text{out}}_N = \bigotimes_{A\in\text{out}(N)} \mathcal{H}_A$, and $p^{\mu_N(\vec{k})}_N$ is the projection $\mathcal{H}^{\text{in}}_N = \bigotimes_{A\in\text{in}(N)} \mathcal{H}_A \to \mathcal{H}^{\text{in}}_{N^{\mu_N(\vec{k})}}$. We will also define the injection $i^{(k_A)_{A\in\text{out}(N)}}_N : \bigotimes_{A\in\text{out}(N)} \mathcal{H}^{k_A}_A \to \mathcal{H}^{\text{out}}_{N^{\mu_N(\vec{k})}}$, and the projection $p^{(k_A)_{A\in\text{in}(N)}}_N : \mathcal{H}^{\text{in}}_{N^{\mu_N(\vec{k})}} \to \bigotimes_{A\in\text{in}(N)} \mathcal{H}^{k_A}_A$: these map $T^{\vec{k}}$'s to the suitable domains and codomains. Note that we then have

$$
\bigotimes_N i^{\mu_N(\vec{k})}_N \circ i^{(k_A)_{A\in\text{out}(N)}}_N = \bigotimes_A i^{k_A}_A , \tag{67a}
$$

$$
\bigotimes_N p^{(k_A)_{A\in\text{out}(N)}}_N \circ p^{\mu_N(\vec{k})}_N = \bigotimes_A p^{k_A}_A , \tag{67b}
$$

where $i^{k_A}_A$ is the injection $\mathcal{H}^{k_A}_A \to \mathcal{H}_A$ and $p^{k_A}_A$ is the projection $\mathcal{H}_A \to \mathcal{H}^{k_A}_A$. Thus,

$$
\begin{aligned}
&\left(\bigotimes_N p_N^{(k_A)_{A\in\mathrm{in}(N)}}\right)_{\mathrm{pad}} \circ T^{\vec{k}} \circ \left(\bigotimes_N i_N^{(k_A)_{A\in\mathrm{out}(N)}}\right)_{\mathrm{pad}} \\
&= \left(\bigotimes_A p_A^{k_A} \circ \pi_A^{k_A} \circ i_A^{k_A}\right) \otimes \left(\bigotimes_{M,\beta\neq\mu_M(\vec{k})} \Theta_{M^\beta}\right) \\
&= \left(\bigotimes_A \mathbb{1}_{A^{k_A}}\right) \otimes \left(\bigotimes_{M,\beta\neq\mu_M(\vec{k})} \Theta_{M^\beta}\right) .
\end{aligned}
\tag{68}
$$

Each of the blocks composing $\mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right]$ is thus unitary once restricted to the suitable subspaces, so $\mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right]$ is unitary as well.

H2 We define, for all branches N^α,

$$
Z^{\mathrm{out}}(N^\alpha) := \Lambda_\Gamma^{-1}\left(\{1\}_{\mathtt{Happens}_{N^\alpha}} \times \bigtimes_{M^\beta\neq N^\alpha} \mathtt{Happens}_{M^\beta}\right) , \tag{69a}
$$

$$
\zeta^{\mathrm{out}}(N^\alpha) := \sum_{(l_{M^\beta})_{M^\beta\in\mathtt{Bran}_\Gamma}\in Z^{\mathrm{out}}(N^\alpha)} \left(\bigotimes_{M^\beta} \pi^{l_{M^\beta}}_{M^\beta_{\mathrm{out}}}\right) , \tag{69b}
$$

and similarly for the Z^{in}'s and ζ^{in}'s. Note that

$$
Z^{\mathrm{out}}(N^\alpha) = \bigsqcup_{\substack{\vec{k}\in\mathtt{PossVal}_\Gamma \\ \mu_N(\vec{k})=\alpha}} (\Lambda_\Gamma^{\mathrm{sec}})^{-1}\left(\vec{k}\right) . \tag{70}
$$

Given their definition, the ζ^{out}'s are commuting orthogonal projectors. Furthermore, as green dashed arrows in $\Gamma^{\mathtt{Bran}}$ represent Λ_Γ's causal structure, we have, for any branch N^α,

$$
Z^{\mathrm{out}}(N^\alpha) := \tilde{Z}^{\mathrm{out}}(N^\alpha) \times \left(\bigtimes_{\exists \text{ no green dashed arrow } M^\beta\to N^\alpha} \mathtt{Ind}^{\mathrm{out}}_{M^\beta}\right) \tag{71}
$$

Through (69b), this implies that $\zeta^{\mathrm{out}}(N^\alpha)$ acts trivially on the M^β's that are not linked to N^α by a green dashed arrow. We can thus in particular see it as the padding of an operator acting only on $\mathcal{P}(N^\alpha)$, or on $\mathcal{P}^{\mathrm{str}}(N^\alpha)$ if N^α is in a red layer. The same applies symmetrically for the ζ^{in}'s. Finally, (65) implies

$$
\begin{aligned}
\mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right] \circ \zeta^{\mathrm{out}}(N^\alpha) &= \sum_{\substack{\vec{k}\in\mathtt{PossVal}_\Gamma \\ \mu_N(\vec{k})=\alpha}} \mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right] \circ \left(\bigotimes_{N\in\mathtt{Nodes}_\Gamma} \pi^{(k_A)_{A\in\mathrm{out}(N)}}_{N^{\mu_N(\vec{k})}_{\mathrm{out}}}\right)_{\mathrm{pad}} \\
&= \sum_{\substack{\vec{k}\in\mathtt{PossVal}_\Gamma \\ \mu_N(\vec{k})=\alpha}} \left(\bigotimes_{N\in\mathtt{Nodes}_\Gamma} \pi^{(k_A)_{A\in\mathrm{out}(N)}}_{N^{\mu_N(\vec{k})}_{\mathrm{out}}}\right)_{\mathrm{pad}} \circ \mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right] \\
&= \zeta^{\mathrm{in}}(N^\alpha) \circ \mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right] ;
\end{aligned}
\tag{72}
$$

thus, $\mathcal{V}_0[\zeta^{\text{out}}(N^\alpha)] = \zeta^{\text{in}}(N^\alpha)$.

H3 We will prove that $Z^{\text{out}}(N^\alpha) \cap \bar{Z}^{\text{out}}(M^\beta)$ is of the form $\tilde{Z} \times \mathtt{Ind}^{\text{out}}_{M^\beta}$, from which (H3) derives. This set can be computed, using bra-ket notations in **Rel**, as $\left(\langle 1|_{\mathtt{Happens}_{N^\alpha}} \otimes \langle 0|_{\mathtt{Happens}_{M^\beta}}\right)_{\text{pad}} \circ \Lambda_\Gamma$. Yet, one can see from the definition of the λ^{aug}'s that

$$\langle 0|_{\mathtt{Happens}_{M^\beta},\text{pad}} \circ \lambda^{\text{aug}}_M = \langle 0|_{\mathtt{Happens}_{M^\beta},\text{pad}} \circ \lambda^{\text{aug}}_M \circ \left|\mathtt{Ind}^{\text{out}}_{M^\beta}\right\rangle\!\left\langle\mathtt{Ind}^{\text{out}}_{M^\beta}\right|_{\mathtt{Ind}^{\text{out}}_{M^\beta},\text{pad}} ; \tag{73}$$

thus,

$$\begin{aligned}
&\left(\langle 1|_{\mathtt{Happens}_{N^\alpha}} \otimes \langle 0|_{\mathtt{Happens}_{M^\beta}}\right)_{\text{pad}} \circ \Lambda_\Gamma \\
&= \left(\langle 1|_{\mathtt{Happens}_{N^\alpha}} \otimes \langle 0|_{\mathtt{Happens}_{M^\beta}}\right)_{\text{pad}} \circ \mathcal{S}^{\mathbf{Rel}}_\Gamma\left[(\lambda^{\text{aug}}_N)_N\right] \\
&= \left(\langle 1|_{\mathtt{Happens}_{N^\alpha}} \otimes \langle 0|_{\mathtt{Happens}_{M^\beta}}\right)_{\text{pad}} \circ \mathcal{S}^{\mathbf{Rel}}_\Gamma\left[(\lambda^{\text{aug}}_N)_{N\neq M} \times \left(\lambda^{\text{aug}}_M \circ \left|\mathtt{Ind}^{\text{out}}_{M^\beta}\right\rangle\!\left\langle\mathtt{Ind}^{\text{out}}_{M^\beta}\right|_{\mathtt{Ind}^{\text{out}}_{M^\beta},\text{pad}}\right)\right] \\
&= \left(\left(\langle 1|_{\mathtt{Happens}_{N^\alpha}} \otimes \langle 0|_{\mathtt{Happens}_{M^\beta}}\right)_{\text{pad}} \circ \mathcal{S}^{\mathbf{Rel}}_\Gamma\left[(\lambda^{\text{aug}}_N)_N\right] \circ |\mathtt{Ind}^{\text{out}}_{M^\beta}\rangle_{\mathtt{Ind}^{\text{out}}_{M^\beta}}\right) \otimes \langle\mathtt{Ind}^{\text{out}}_{M^\beta}|_{\mathtt{Ind}^{\text{out}}_{M^\beta}} ,
\end{aligned} \tag{74}$$

which shows that indeed $Z^{\text{out}}(N^\alpha) \cap \bar{Z}^{\text{out}}(M^\beta) = \tilde{Z} \times \mathtt{Ind}^{\text{out}}_{M^\beta}$. The proof for the Z^{in}'s is symmetric.

H4 (H4) comes from the fact that, for $\alpha \neq \beta$, one has $Z^{\text{out}}(N^\alpha) \cap Z^{\text{out}}(N^\beta) = \emptyset$, which can be derived directly from (70).

H5 We take $Q = \{N^{\alpha(N)} \mid N \in \tilde{Q}\} \subseteq \mathtt{Bran}_\Gamma$. Then,

$$\begin{aligned}
&\mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_N\right] \circ \prod_{N\in\tilde{Q}} \zeta^{\text{out}}(N^{\alpha(N)}) \\
&\overset{(70)}{=} \sum_{\substack{\vec{k}\in\mathtt{PossVal}_\Gamma \\ \forall N\in\tilde{Q},\mu_N(\vec{k})=\alpha(N)}} \mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_N\right] \circ \left(\bigotimes_{N\in\mathtt{Nodes}_\Gamma} \pi^{(k_A)_{A\in\text{out}(N)}}_{N^{\mu_N(\vec{k})}_{\text{out}}}\right)_{\text{pad}} \\
&\overset{(62)}{=} \sum_{\substack{\vec{k}\in\mathtt{PossVal}_\Gamma \\ \forall N\in\tilde{Q},\mu_N(\vec{k})=\alpha(N)}} \mathcal{S}_{\text{pad}}\left[\left(\left(\bigotimes_{A\in\text{out}(N)} \pi^{k_A}_A\right)_{\text{pad}} \circ \text{EXCH}_N\right)_N\right] \\
&= \mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_{N\notin\tilde{Q}} \times \left(\sum_{(k_A)_{A\in\text{out}(N)}\in\mathtt{Ind}^{\text{out}}_{N^\alpha}} \left(\bigotimes_{A\in\text{out}(N)} \pi^{k_A}_A\right)_{\text{pad}} \circ \text{EXCH}_N\right)_{N\in\tilde{Q}}\right] \\
&= \mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_{N\notin\tilde{Q}} \times (\pi^\alpha_{N^{\text{out}}} \circ \text{EXCH}_N)_{N\in\tilde{Q}}\right] \\
&\overset{(41)}{=} \mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_{N\notin\tilde{Q}} \times \left(V^\alpha_{0,N}\right)_{N\in\tilde{Q}}\right] .
\end{aligned} \tag{75}$$

H6 We take N^α in a green layer, and $f \in \mathrm{Lin}\left[\mathcal{H}_{N^\alpha_{\mathrm{in}}}\right]$. We then have (note that $\zeta^{\mathrm{in}}(N^\alpha)$ doesn't act on N^α_{in})

$$\begin{aligned}
&\mathcal{V}_0^\dagger\left[f_{\mathrm{pad}}\right] \circ \zeta^{\mathrm{out}}_{\mathrm{pad}}(N^\alpha) = \mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right]^\dagger \circ f_{\mathrm{pad}} \circ \mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right] \circ \zeta^{\mathrm{out}}_{\mathrm{pad}}(N^\alpha) \\
&\overset{(72)}{=} \mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right]^\dagger \circ f_{\mathrm{pad}} \circ \zeta^{\mathrm{in}}_{\mathrm{pad}}(N^\alpha) \circ \mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right] \circ \zeta^{\mathrm{out}}_{\mathrm{pad}}(N^\alpha) \\
&= \mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right]^\dagger \circ \zeta^{\mathrm{in}}_{\mathrm{pad}}(N^\alpha) \circ f_{\mathrm{pad}} \circ \mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right] \circ \zeta^{\mathrm{out}}_{\mathrm{pad}}(N^\alpha) \\
&\overset{(72)}{=} \zeta^{\mathrm{out}}_{\mathrm{pad}}(N^\alpha) \circ \mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right]^\dagger \circ f_{\mathrm{pad}} \circ \mathcal{S}_{\mathrm{pad}}\left[(\mathrm{EXCH}_N)_N\right] \circ \zeta^{\mathrm{out}}_{\mathrm{pad}}(N^\alpha) \\
&\overset{(69b),(70)}{=} \sum_{\substack{\vec{k},\vec{l}\in\mathtt{PossVal}_\Gamma \\ \mu_N(\vec{k})=\mu_N(\vec{l})=\alpha}} \left(T^{\vec{l}}\right)^\dagger \circ f_{\mathrm{pad}} \circ T^{\vec{k}}\,.
\end{aligned} \tag{76}$$

Furthermore, taking $M^\beta \notin \mathcal{P}^{\mathrm{str}}(N^\alpha)$, because N^α is in a green layer we know that there is no red dashed arrow from M^β to N^α, and thus $\zeta^{\mathrm{in}}(M^\beta)$ doesn't act on N^α_{in}. We can thus apply the same computation to it as well, which leads to

$$\begin{aligned}
&\mathcal{V}_0^\dagger\left[f_{\mathrm{pad}}\right] \circ \zeta^{\mathrm{out}}_{\mathrm{pad}}(N^\alpha) = \mathcal{V}_0\left[f_{\mathrm{pad}}\right] \circ \zeta^{\mathrm{out}}_{\mathrm{pad}}(N^\alpha) \circ \left(\zeta^{\mathrm{out}}_{\mathrm{pad}}(M^\beta) + \bar{\zeta}^{\mathrm{out}}_{\mathrm{pad}}(M^\beta)\right) \\
&= \zeta^{\mathrm{out}}_{\mathrm{pad}}(N^\alpha) \circ \zeta^{\mathrm{out}}_{\mathrm{pad}}(M^\beta) \circ \mathcal{V}_0\left[f_{\mathrm{pad}}\right] \circ \zeta^{\mathrm{out}}_{\mathrm{pad}}(N^\alpha) \circ \zeta^{\mathrm{out}}_{\mathrm{pad}}(M^\beta) \\
&+ \zeta^{\mathrm{out}}_{\mathrm{pad}}(N^\alpha) \circ \bar{\zeta}^{\mathrm{out}}_{\mathrm{pad}}(M^\beta) \circ \mathcal{V}_0\left[f_{\mathrm{pad}}\right] \circ \zeta^{\mathrm{out}}_{\mathrm{pad}}(N^\alpha) \circ \bar{\zeta}^{\mathrm{out}}_{\mathrm{pad}}(M^\beta) \\
&= \sum_{\substack{\vec{k},\vec{l}\in\mathtt{PossVal}_\Gamma \\ \mu_N(\vec{k})=\mu_N(\vec{l})=\alpha \\ \mu_M(\vec{k})=\beta \iff \mu_M(\vec{l})=\beta}} \left(T^{\vec{l}}\right)^\dagger \circ f_{\mathrm{pad}} \circ T^{\vec{k}}\,;
\end{aligned} \tag{77}$$

in other words, in the sum above, the values of $\vec{k}$ and $\vec{l}$ that lead to attributing different statuses to M^β correspond to null terms, so that one can skip them in the summation. More generally, one can apply this reasoning to all branches $M^\beta \notin \mathcal{P}^{\mathrm{str}}(N^\alpha)$, leading to

$$\mathcal{V}_0\left[f_{\mathrm{pad}}\right] \circ \zeta^{\mathrm{out}}_{\mathrm{pad}}(N^\alpha) = \sum_{\substack{\vec{k},\vec{l}\in\mathtt{PossVal}_\Gamma \\ \mu_N(\vec{k})=\mu_N(\vec{l})=\alpha \\ \forall M^\beta\notin\mathcal{P}^{\mathrm{str}}(N^\alpha),\mu_M(\vec{k})=\beta \iff \mu_M(\vec{l})=\beta}} \left(T^{\vec{l}}\right)^\dagger \circ f_{\mathrm{pad}} \circ T^{\vec{k}}\,. \tag{78}$$

Using (68), we rewrite $T^{\vec{k}}$, for an arbitrary $\vec{k}$, as

$$\begin{aligned}
T^{\vec{k}} = &\left(\bigotimes_M p_M^{(k_A)_{A\in\mathrm{in}(M)}}\right)^\dagger_{\mathrm{pad}} \circ \left(\left(\bigotimes_A \mathbb{1}_{A^{k_A}}\right) \otimes \left(\bigotimes_{M,\beta\neq\mu_M(\vec{k})} \Theta_{M^\beta}\right)\right) \\
&\circ \left(\bigotimes_M i_M^{(k_A)_{A\in\mathrm{out}(M)}}\right)^\dagger_{\mathrm{pad}}\,.
\end{aligned} \tag{79}$$

Now, we take $\vec{k},\vec{l} \in \mathtt{PossVal}_\Gamma$ satisfying the requirements we pinned down earlier; we can then compute

$$
\begin{aligned}
&T^{\vec{l}^\dagger} \circ f_{\text{pad}} \circ T^{\vec{k}} \\
&= \left(\bigotimes_M i_M^{(l_A)_{A\in\text{out}(M)}}\right)_{\text{pad}} \circ \left(\left(\bigotimes_A \mathbb{1}_{A^{l_A}}\right) \otimes \left(\bigotimes_{M,\beta\neq\mu_M(\vec{l})} \Theta^\dagger_{M^\beta}\right)\right) \\
&\circ \left(\bigotimes_M p_M^{(l_A)_{A\in\text{in}(M)}}\right)_{\text{pad}} \circ f_{\text{pad}} \circ \left(\bigotimes_M p_M^{(k_A)_{A\in\text{in}(M)}}\right)^\dagger_{\text{pad}} \\
&\circ \left(\left(\bigotimes_A \mathbb{1}_{A^{k_A}}\right) \otimes \left(\bigotimes_{M,\beta\neq\mu_M(\vec{k})} \Theta_{M^\beta}\right)\right) \circ \left(\bigotimes_M i_M^{(k_A)_{A\in\text{out}(M)}}\right)^\dagger_{\text{pad}} \\
&= \left(\bigotimes_M i_M^{(l_A)_{A\in\text{out}(M)}}\right)_{\text{pad}} \circ \left[\left(p_N^{(l_A)_{A\in\text{in}(N)}} \circ f \circ \left(p_N^{(k_A)_{A\in\text{in}(N)}}\right)^\dagger\right)\right. \\
&\otimes \left(\bigotimes_{\substack{M\neq N\\ \mu_M(\vec{l})\neq\mu_M(\vec{k})}} \left(\Theta^\dagger_{M^{\mu_M(\vec{k})}} \circ \left(p_M^{(k_A)_{A\in\text{in}(M)}}\right)^\dagger\right) \otimes \left(p_M^{(l_A)_{A\in\text{in}(M)}} \circ \Theta_{M^{\mu_M(\vec{l})}}\right)\right) \\
&\left.\otimes \left(\bigotimes_{\substack{M\neq N\\ \mu_M(\vec{l})=\mu_M(\vec{k})=1}} p_M^{(l_A)_{A\in\text{in}(M)}} \circ \left(p_M^{(k_A)_{A\in\text{in}(M)}}\right)^\dagger\right)\right]_{\text{pad}} \circ \left(\bigotimes_M i_M^{(k_A)_{A\in\text{out}(M)}}\right)^\dagger_{\text{pad}} .
\end{aligned}
\tag{80}
$$

Note that each of the $p_M^{(l_A)_{A\in\text{in}(M)}} \circ \left(p_M^{(k_A)_{A\in\text{in}(M)}}\right)^\dagger$ terms, for M such that $M^{\mu_M(\vec{k})} \notin \mathcal{P}^{\text{str}}(N^\alpha) \cup \{N^\alpha\}$, can be rewritten as $\bigotimes_{A\in\text{in}(M)} p_A^{l_A} \circ i_A^{k_A}$ (which is the identity if $k_A = l_A \forall A \in \text{in}(M)$, and zero otherwise).

Now, for any M such that $M^{\mu_M(\vec{k})} \notin \mathcal{P}^{\text{str}}(N^\alpha)$, and for any O such that $O^{\mu_O(\vec{k})} \in \mathcal{P}^{\text{str}}(N^\alpha) \cup \{N^\alpha\}$, there is no arrow $A \in \texttt{Link}(M, O)$ such that $\left|A^{k_A}\right| \neq 1$, as that would imply the existence of a solid arrow from $M^{\mu_M(\vec{k})}$ to $O^{\mu_O(\vec{k})}$, which would contradict $M^{\mu_M(\vec{k})} \notin \mathcal{P}^{\text{str}}(N^\alpha)$. Thus, all of the non-trivial arrows in $\text{out}(M)$ go to O's such that $O \neq N$ and $\mu_O(\vec{l}) = \mu_O(\vec{k})$. Thus this implies that, if one doesn't have $k_A = l_A \forall A \in \text{out}(M)$ then the whole expression is null; while otherwise, the term in square brackets acts trivially on each of the $A \in \text{out}(M)$ – in other words, the arrows coming out of M are never acted on and simply link $i_M^{(k_A)_{A\in\text{out}(M)}\dagger}$ directly to $i_M^{(l_A)_{A\in\text{out}(M)}}$. One can thus reorganise this expression (neglecting the existence of all the trivial spaces) as

$$
\begin{aligned}
T^{\vec{l}^\dagger} \circ f_{\text{pad}} \circ T^{\vec{k}} = \Bigg[& \left(\bigotimes_{\substack{M \\ M^{\mu_M(\vec{l})} \in \mathcal{P}^{\text{str}}(N^\alpha)}} i_M^{(l_A)_{A \in \text{out}(M)}} \right)_{\text{pad}} \circ \left[\left(p_N^{(l_A)_{A \in \text{in}(N)}} \circ f \circ \left(p_N^{(k_A)_{A \in \text{in}(N)}} \right)^\dagger \right) \right. \\
& \otimes \left(\bigotimes_{\substack{M \\ M^{\mu_M(\vec{l})}, M^{\mu_M(\vec{k})} \in \mathcal{P}^{\text{str}}(N^\alpha) \\ \mu_M(\vec{l}) \neq \mu_M(\vec{k})}} \left(\Theta^\dagger_{M^{\mu_M(\vec{k})}} \circ \left(p_M^{(k_A)_{A \in \text{in}(M)}} \right)^\dagger \right) \otimes \left(p_M^{(l_A)_{A \in \text{in}(M)}} \circ \Theta_{M^{\mu_M(\vec{l})}} \right) \right) \\
& \otimes \left(\bigotimes_{\substack{M \\ M^{\mu_M(\vec{l})}, M^{\mu_M(\vec{k})} \in \mathcal{P}^{\text{str}}(N^\alpha) \\ \mu_M(\vec{l}) = \mu_M(\vec{k})}} \bigotimes_{A \in \text{in}(M)} p_A^{l_A} \circ i_A^{k_A} \right) \otimes \left. \left(\bigotimes_{\substack{O \\ O^{\mu_O(\vec{k})} \notin \mathcal{P}^{\text{str}}(N^\alpha)}} \bigotimes_{\substack{M \\ M^{\mu_M(\vec{k})} \in \mathcal{P}^{\text{str}}(N^\alpha)}} \bigotimes_{A \in \texttt{Link}(M,O)} p_A^{l_A} \circ i_A^{k_A} \right) \right] \\
& \circ \left(\bigotimes_{\substack{M \\ M^{\mu_M(\vec{k})} \in \mathcal{P}^{\text{str}}(N^\alpha)}} i_M^{(k_A)_{A \in \text{out}(M)}} \right)^\dagger_{\text{pad}} \Bigg]_{\text{pad}} \otimes \left(\left(\bigotimes_{\substack{M \\ M^{\mu_M(\vec{k})} \notin \mathcal{P}^{\text{str}}(N^\alpha)}} \pi^{(k_A)_{A \in \text{out}(M)}}_{M^{\mu_M(\vec{k})}_{\text{out}}} \right) \otimes \left(\bigotimes_{\substack{M, \beta \neq \mu_M(\vec{k}) \\ M^\beta \notin \mathcal{P}^{\text{str}}(N^\alpha)}} \mathbb{1}_{M^\beta_{\text{out}}} \right) \right) ;
\end{aligned}
\tag{81}
$$

note how the action on the M^β_{out}'s for $M^\beta \notin \mathcal{P}^{\text{str}}(N^\alpha)$ now only consists of a projector independent of f. Note that in the second bracket of the third line, the condition $M^{\mu_M(\vec{k})} \in \mathcal{P}^{\text{str}}(N^\alpha)$ could equivalently have been replaced with $M^{\mu_M(\vec{l})} \in \mathcal{P}^{\text{str}}(N^\alpha)$, because we know that $M^{\mu_M(\vec{k})} \notin \mathcal{P}^{\text{str}}(N^\alpha) \implies M^{\mu_M(\vec{l})} = M^{\mu_M(\vec{k})} \notin \mathcal{P}^{\text{str}}(N^\alpha)$, and conversely; so that we have the equivalence $M^{\mu_M(\vec{k})} \notin \mathcal{P}^{\text{str}}(N^\alpha) \iff M^{\mu_M(\vec{l})} \notin \mathcal{P}^{\text{str}}(N^\alpha)$.

Both bracketed terms in the third line can be rewritten simply as Kronecker deltas, of the form:

$$
\left(\prod_{\substack{M \\ M^{\mu_M(\vec{l})}, M^{\mu_M(\vec{k})} \notin \mathcal{P}^{\text{str}}(N^\alpha) \\ \mu_M(\vec{l}) = \mu_M(\vec{k})}} \prod_{A \in \text{in}(M)} \delta_{k_A, l_A} \right) \cdot \left(\prod_{\substack{O \\ O^{\mu_O(\vec{k})} \notin \mathcal{P}^{\text{str}}(N^\alpha)}} \prod_{\substack{M \\ M^{\mu_M(\vec{k})} \in \mathcal{P}^{\text{str}}(N^\alpha)}} \prod_{A \in \texttt{Link}(M,O)} \delta_{k_A, l_A} \right) .
\tag{82}
$$

We now take $\vec{q} = (q^{O^\gamma})_{O^\gamma \in \texttt{Bran}_\Gamma}, \vec{r} = (r^{O^\gamma})_{O^\gamma \in \texttt{Bran}_\Gamma} \in \bigtimes_{O^\gamma \in \texttt{Bran}_\Gamma} \texttt{Ind}^{\text{out}}_{O^\gamma}$. We then have

$$
\begin{aligned}
&\left(\bigotimes_{O^\gamma\in\mathtt{Bran}_\Gamma}\pi^{r^{O^\gamma}}_{O^\gamma_{\text{out}}}\right)\circ\mathcal{V}_0^\dagger[f_{\text{pad}}]\circ\zeta^{\text{out}}_{\text{pad}}(N^\alpha)\circ\left(\bigotimes_{O^\gamma\in\mathtt{Bran}_\Gamma}\pi^{q^{O^\gamma}}_{O^\gamma_{\text{out}}}\right)\\
&=\left(\bigotimes_{O^\gamma\in\mathtt{Bran}_\Gamma}\pi^{r^{O^\gamma}}_{O^\gamma_{\text{out}}}\right)\circ\mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_N\right]^\dagger\circ f_{\text{pad}}\circ\mathcal{S}_{\text{pad}}\left[(\text{EXCH}_N)_N\right]\circ\left(\bigotimes_{O^\gamma\in\mathtt{Bran}_\Gamma}\pi^{q^{O^\gamma}}_{O^\gamma_{\text{out}}}\right)\circ\zeta^{\text{out}}_{\text{pad}}(N^\alpha)\\
&=\left(\bigotimes_{O^\gamma\in\mathtt{Bran}_\Gamma}\pi^{r^{O^\gamma}}_{O^\gamma_{\text{out}}}\right)\circ\left(T^{\Lambda^{\text{sec}}_\Gamma(\vec{r})}\right)^\dagger\circ f_{\text{pad}}\circ T^{\Lambda^{\text{sec}}_\Gamma(\vec{q})}\circ\left(\bigotimes_{O^\gamma\in\mathtt{Bran}_\Gamma}\pi^{q^{O^\gamma}}_{O^\gamma_{\text{out}}}\right)\circ\zeta^{\text{out}}_{\text{pad}}(N^\alpha)\,.
\end{aligned}
\tag{83}
$$

Note that $\mu_M\circ\Lambda^{\text{sec}}_\Gamma(\vec{q})$ denotes the only branch β of M such that the M^β term of $\Lambda_\Gamma(\vec{q})$, which we denote $\Lambda^{M^\beta}_\Gamma(\vec{q})$, is 1. By the previous considerations, the term above can thus be non null only if $\Lambda^{N^\alpha}_\Gamma(\vec{q})=\Lambda^{N^\alpha}_\Gamma(\vec{r})=1$ and if $\Lambda^{M^\beta}_\Gamma(\vec{q})=\Lambda^{M^\beta}_\Gamma(\vec{r})\,\forall M^\beta\notin\mathcal{P}^{\text{str}}(N^\alpha)$. Furthermore, when this is the case, then we can use (81) and get

$$
\begin{aligned}
&\left(\bigotimes_{O^\gamma\in\mathtt{Bran}_\Gamma}\pi^{r^{O^\gamma}}_{O^\gamma_{\text{out}}}\right)\circ\mathcal{V}_0^\dagger[f_{\text{pad}}]\circ\zeta^{\text{out}}_{\text{pad}}(N^\alpha)\circ\left(\bigotimes_{O^\gamma\in\mathtt{Bran}_\Gamma}\pi^{q^{O^\gamma}}_{O^\gamma_{\text{out}}}\right)\\
&=F^{(r^{O^\gamma})_{O^\gamma\in\mathcal{P}^{\text{str}}(N^\alpha)},(q^{O^\gamma})_{O^\gamma\in\mathcal{P}^{\text{str}}(N^\alpha)}}\otimes\left(\bigotimes_{M^\beta\notin\mathcal{P}^{\text{str}}(N^\alpha)}\pi^{q^{M^\beta}}_{M^\beta_{\text{out}}}\right)\\
&\cdot\prod_{O^\gamma\notin\mathcal{P}^{\text{str}}(N^\alpha)}\delta_{q^{O^\gamma},r^{O^\gamma}}\,,
\end{aligned}
\tag{84}
$$

where

$$
\begin{aligned}
&F^{(r^{O^\gamma})_{O^\gamma\in\mathcal{P}^{\mathrm{str}}(N^\alpha)},(q^{O^\gamma})_{O^\gamma\in\mathcal{P}^{\mathrm{str}}(N^\alpha)}} \\
&= \left(\bigotimes_{O^\gamma\in\mathcal{P}^{\mathrm{str}}(N^\alpha)} \pi^{r^{O^\gamma}}_{O^\gamma_{\mathrm{out}}}\right) \circ \left(\bigotimes_{\substack{M^\beta\in\mathcal{P}^{\mathrm{str}}(N^\alpha)\\ \Lambda_\Gamma^{M^\beta}(\vec{r})=1}} i_M^{(l_A)_{A\in\mathrm{out}(M)}}\right)_{\mathrm{pad}} \circ \left[\left(p_N^{(l_A)_{A\in\mathrm{in}(N)}} \circ f \circ \left(p_N^{(k_A)_{A\in\mathrm{in}(N)}}\right)^\dagger\right)\right. \\
&\left.\otimes \left(\bigotimes_{\substack{M \text{ s.t. } \exists\beta,\beta':\\ M^\beta,M^{\beta'}\in\mathcal{P}^{\mathrm{str}}(N^\alpha)\\ \Lambda_\Gamma^{M^\beta}(\vec{q})=\Lambda_\Gamma^{M^{\beta'}}(\vec{r})=1}} \left(\Theta^\dagger_{M^\beta} \circ \left(p_M^{(k_A)_{A\in\mathrm{in}(M)}}\right)^\dagger\right) \otimes \left(p_M^{(l_A)_{A\in\mathrm{in}(M)}} \circ \Theta_{M^{\beta'}}\right)\right)\right] \\
&\circ \left(\bigotimes_{\substack{M^\beta\in\mathcal{P}^{\mathrm{str}}(N^\alpha)\\ \Lambda_\Gamma^{M^\beta}(\vec{q})=1}} i_M^{(k_A)_{A\in\mathrm{out}(M)}}\right)^\dagger_{\mathrm{pad}} \circ \left(\bigotimes_{O^\gamma\in\mathcal{P}^{\mathrm{str}}(N^\alpha)} \pi^{q^{O^\gamma}}_{O^\gamma_{\mathrm{out}}}\right) \cdot \left(\prod_{\substack{M^\beta\in\mathcal{P}^{\mathrm{str}}(N^\alpha)\\ \Lambda_\Gamma^{M^\beta}(\vec{r})=\Lambda_\Gamma^{M^\beta}(\vec{q})=1}} \prod_{A\in\mathrm{in}(M)} \delta_{k_A,l_A}\right) \\
&\cdot \left(\prod_{\substack{O\\ \sum_{O^\gamma\in\mathcal{P}^{\mathrm{str}}(N^\alpha)}\Lambda_\Gamma^{O^\gamma}(\vec{q})=0}} \prod_{\substack{M\\ \sum_{M^\beta\in\mathcal{P}^{\mathrm{str}}(N^\alpha)}\Lambda_\Gamma^{M^\beta}(\vec{q})=1}} \prod_{A\in\mathtt{Link}(M,O)} \delta_{k_A,l_A}\right),
\end{aligned}
\tag{85}
$$

with $\vec{k} := \Lambda_\Gamma^{\mathrm{sec}}(\vec{q})$, $\vec{l} := \Lambda_\Gamma^{\mathrm{sec}}(\vec{r})$. Note how $F^{(r^{O^\gamma})_{O^\gamma\in\mathcal{P}^{\mathrm{str}}(N^\alpha)},(q^{O^\gamma})_{O^\gamma\in\mathcal{P}^{\mathrm{str}}(N^\alpha)}}$, which comes from the term in square brackets in (81) and acts only on P and on $\mathcal{P}^{\mathrm{str}}(N^\alpha)$, doesn't depend on the value of $(q^{O^\gamma})_{O^\gamma\notin\mathcal{P}^{\mathrm{str}}(N^\alpha)}$: indeed, the values of $\Lambda_\Gamma^{M^\beta}(\vec{q})$ and of $\Lambda_\Gamma^{M^\beta}(\vec{r})$ don't depend on the values of the q^{O^γ} for $O^\gamma \notin \mathcal{P}^{\mathrm{str}}(N^\alpha)$, as there are no green dashed arrows from these O^γ's to any $M^\beta \in \mathcal{P}^{\mathrm{str}}(N^\alpha)$. Similarly, the relevant values of the k_A's are those such that $A \in \mathrm{out}(M)$ for some $M^\beta \in \mathcal{P}^{\mathrm{str}}(N^\alpha)$ such that $\Lambda_\Gamma^{M^\beta}(\vec{q}) = 1$; thus they are just equal to the A-value of q^{M^β}. The same goes for the l_A's.

So far, we proved that (84) holds for $\vec{q}$ and $\vec{r}$ satisfying: $\Lambda_\Gamma^{N^\alpha}(\vec{q}) = \Lambda_\Gamma^{N^\alpha}(\vec{r}) = 1$ and $\Lambda_\Gamma^{M^\beta}(\vec{q}) = \Lambda_\Gamma^{M^\beta}(\vec{r})\,\forall M^\beta \notin \mathcal{P}^{\mathrm{str}}(N^\alpha)$. We now want to prove that the same holds when the latter condition is not satisfied – or in other words, that in this case the RHS is also null. We will thus prove that if $\Lambda_\Gamma^{M^\beta}(\vec{q}) = \Lambda_\Gamma^{M^\beta}(\vec{r})\,\forall M^\beta \notin \mathcal{P}^{\mathrm{str}}(N^\alpha)$ does not hold, then (85) is null, and thus the RHS in (84) is null as well.

We suppose (85) is not null, and take $M^\beta \notin \mathcal{P}^{\mathrm{str}}(N^\alpha)$ such that $\Lambda_\Gamma^{M^\beta}(\vec{q}) = 1$. Taking $A \in \mathrm{out}(M)$, and denoting $O := \mathtt{head}(A)$ and γ such that $\Lambda_\Gamma^{O^\gamma}(\vec{q}) = 1$, we have: either $O^\gamma \in \mathcal{P}^{\mathrm{str}}(N^\alpha)$, in which case $k_A = l_A$ by the penultimate term in (85); or $O^\gamma \notin \mathcal{P}^{\mathrm{str}}(N^\alpha)$, in which case $k_A = l_A$ by the last term in (85). Thus we have $k_A = l_A \forall A \in \mathrm{out}(M)$, and thus $\Lambda_\Gamma^{M^\beta}(\vec{r}) = 1$. Symmetrically, $\Lambda_\Gamma^{M^\beta}(\vec{r}) = 1$ implies $\Lambda_\Gamma^{M^\beta}(\vec{q}) = 1$, so we indeed get $\Lambda_\Gamma^{M^\beta}(\vec{r}) = \Lambda_\Gamma^{M^\beta}(\vec{q})$.

Therefore, (84) holds for $\vec{q}$ satisfying $\Lambda_\Gamma^{N^\alpha}(\vec{q}) = 1$. We can thus finally compute

$$
\begin{aligned}
&\mathcal{V}_0^\dagger[f_{\text{pad}}] \circ \zeta_{\text{pad}}^{\text{out}}(N^\alpha) \\
&= \sum_{\substack{\vec{q},\vec{r} \\ \Lambda_\Gamma^{N^\alpha}(\vec{q})=1}} \left(\bigotimes_{O^\gamma \in \mathtt{Bran}_\Gamma} \pi_{O^\gamma_{\text{out}}}^{r^{O^\gamma}} \right) \circ \mathcal{V}_0^\dagger[f_{\text{pad}}] \circ \zeta_{\text{pad}}^{\text{out}}(N^\alpha) \circ \left(\bigotimes_{O^\gamma \in \mathtt{Bran}_\Gamma} \pi_{O^\gamma_{\text{out}}}^{q^{O^\gamma}} \right) \\
&\overset{(84)}{=} \sum_{\substack{\vec{q},\vec{r} \\ \Lambda_\Gamma^{N^\alpha}(\vec{q})=1}} F^{(r^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)}, (q^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)}} \\
&\otimes \left(\bigotimes_{M^\beta \notin \mathcal{P}^{\text{str}}(N^\alpha)} \pi_{M^\beta_{\text{out}}}^{q^{M^\beta}} \right) \cdot \prod_{O^\gamma \notin \mathcal{P}^{\text{str}}(N^\alpha)} \delta_{q^{O^\gamma}, r^{O^\gamma}} \\
&= \sum_{\substack{(q^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)} \\ (r^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)}}} \left(F^{(r^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)}, (q^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)}} \otimes \sum_{\substack{(q^{O^\gamma})_{O^\gamma \notin \mathcal{P}^{\text{str}}(N^\alpha)} \\ \Lambda_\Gamma^{N^\alpha}(\vec{q})=1}} \left(\bigotimes_{M^\beta \notin \mathcal{P}^{\text{str}}(N^\alpha)} \pi_{M^\beta_{\text{out}}}^{q^{M^\beta}} \right) \right) \\
&= \left(\sum_{\substack{(q^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)} \\ (r^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)}}} F^{(r^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)}, (q^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)}} \right) \circ \left(\sum_{\substack{(s^{O^\gamma})_{O^\gamma \in \mathtt{Bran}_\Gamma} \\ \Lambda_\Gamma^{N^\alpha}(\vec{s})=1}} \bigotimes_{M^\beta \notin \mathcal{P}^{\text{str}}(N^\alpha)} \pi_{M^\beta_{\text{out}}}^{s^{M^\beta}} \right)_{\text{pad}} \\
&= \left(\sum_{\substack{(q^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)} \\ (r^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)}}} F^{(r^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)}, (q^{O^\gamma})_{O^\gamma \in \mathcal{P}^{\text{str}}(N^\alpha)}} \right) \circ \zeta_{\text{pad}}^{\text{out}}(N^\alpha)\,;
\end{aligned}
\tag{86}
$$

denoting the left-hand factor as f' yields (58).

H7 The proof of (H7) is the symmetric of that of (H6).

B.4.4 Proof of the induction step

We suppose the induction hypotheses are all satisfied up to step i. We write $M^\beta := B(i+1)$, the branch we have to refill in this induction step.

We first consider the case: neither i nor $i+1$ are special steps. Note that, because the branches have been ordered so that all branches of a same layer are next to each other, the fact that $i+1$ is not a special step entails: M^β is in a red layer $\implies \mathcal{P}_i(M^\beta) = \emptyset$.

H1 Let us first prove H1 at step $i+1$. From (53) and (H5) applied to $Q = \{M^\beta\}$, we have

$$
\begin{aligned}
\mathcal{S}_{\text{pad}}[(V_{N,i})_N] = {} & \zeta_{i,\text{pad}}^{\text{in}}(M^\beta) \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_{N \neq M} \times (V_{M,i}^\beta)] \circ \zeta_{i,\text{pad}}^{\text{out}}(M^\beta) \\
& + \bar{\zeta}_{i,\text{pad}}^{\text{in}}(M^\beta) \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_{N \neq M} \times (\bar{V}_{M,i}^\beta)] \circ \bar{\zeta}_{i,\text{pad}}^{\text{out}}(M^\beta)\,.
\end{aligned}
\tag{87}
$$

Furthermore,

$$\mathcal{S}_{\text{pad}}[(V_{N,i})_{N\neq M}\times(V^{\beta}_{M,i+1})] = \text{Tr}_{M^{\beta}_{\text{out}}}[U^{\beta}_{\text{pad}}\circ\mathcal{S}_{\text{pad}}[(V_{N,i})_{N\neq M}\times(V^{\beta}_{M,i})]]\,, \tag{88a}$$

$$\mathcal{S}_{\text{pad}}[(V_{N,i})_{N\neq M}\times(\bar{V}^{\beta}_{m,i+1})] = \frac{1}{\texttt{dim}(M^{\beta})}\,\text{Tr}_{M^{\beta}_{\text{out}}}[\Theta^{\dagger}_{M^{\beta},\text{pad}}\circ\mathcal{S}_{\text{pad}}[(V_{N,i})_{N\neq M}\times(\bar{V}^{\beta}_{M,i})]]\,, \tag{88b}$$

so, because $\zeta^{\text{in}}_{i}(M^{\beta})$ doesn't act on M^{β}_{in}, and $\zeta^{\text{out}}_{i}(M^{\beta})$ doesn't act on M^{β}_{out},

$$\begin{aligned}\mathcal{S}_{\text{pad}}[(V_{N,i+1})_N] &= \mathcal{S}_{\text{pad}}[(V_{N,i})_{N\neq M}\times(V^{\beta}_{M,i+1})] + \mathcal{S}_{\text{pad}}[(V_{N,i})_{N\neq M}\times(\bar{V}^{\beta}_{M,i+1})]\\ &= \zeta^{\text{in}}_{i,\text{pad}}(M^{\beta})\circ\text{Tr}_{M^{\beta}_{\text{out}}}[U^{\beta}_{\text{pad}}\circ\mathcal{S}_{\text{pad}}[(V_{N,i})_{N\neq M}\times(V^{\beta}_{M,i})]]\circ\zeta^{\text{out}}_{i,\text{pad}}(M^{\beta})\\ &+ \bar{\zeta}^{\text{in}}_{i,\text{pad}}(M^{\beta})\circ\frac{1}{\texttt{dim}(M^{\beta})}\,\text{Tr}_{M^{\beta}_{\text{out}}}[\Theta^{\dagger}_{M^{\beta},\text{pad}}\circ\mathcal{S}_{\text{pad}}[(V_{N,i})_{N\neq M}\times(\bar{V}^{\beta}_{M,i})]]\circ\bar{\zeta}^{\text{out}}_{i,\text{pad}}(M^{\beta})\,.\end{aligned} \tag{89}$$

Therefore, $\mathcal{S}_{\text{pad}}[(V_{N,i+1})_N]$ can be decomposed into two terms: one that can be considered as a linear map from the subspace of $\mathcal{H}^{\text{out}}_{i}$ defined by the projector $\zeta^{\text{out}}_{i,\text{pad}}(M^{\beta})$, to the subspace of $\mathcal{H}^{\text{in}}_{i}$ defined by the projector $\zeta^{\text{in}}_{i,\text{pad}}(M^{\beta})$; and one that can be considered a linear map from, and to, the subspaces orthogonal to these. We now have to prove that each of these two terms is unitary.

We start with the first term. (H6) implies that $U^{\beta}_{\text{pad}}\circ\mathcal{S}_{\text{pad}}[(V_{N,i})_{N\neq M}\times(V^{\beta}_{M,i})]$ features no causal influence from M^{β}_{out} to M^{β}_{in}, via the characterisation of causal influence in terms of algebras [38] (note that it makes sense to talk about the factors M^{β}_{out} and M^{β}_{in} of its input and output spaces because the $\zeta(M^{\beta})$'s do not act on these). Therefore, one can find a unitary causal decomposition of it as $W^2\circ(\text{SWAP}_{M^{\beta}_{\text{out}},M^{\beta}_{\text{in}}}\otimes\mathbb{1})\circ W^1$, where W^1 doesn't act on M^{β}_{out} and W^2 doesn't act on M^{β}_{in}. The first term in (89) – with its input and output spaces suitably restricted – is thus of the form $W^2\circ(U^{\beta}\otimes\mathbb{1})\circ W^1$, which is unitary.

As for the second term, one can see from the definition of the $V^{\alpha}_{M,i}$'s that $\mathcal{S}_{\text{pad}}[(V_{N,i})_{N\neq M}\times(\bar{V}^{\beta}_{M,i})]$ is of the form $\Theta_{M^{\beta}}\otimes W$, with W a unitary (once restricted to the suitable subspaces). Therefore, the term can simply be rewritten as W.

We have therefore proven (H1) at rank $i+1$.

A Lemma. Before turning to the other induction hypotheses, we prove a Lemma that we will need to use a few times to compute how $\mathcal{V}^{\dagger}_{i+1}$ acts on sufficiently well-behaved linear operators.

Lemma B.5. *Let $g\in\text{Lin}[\mathcal{H}^{\text{in}}_{i}]$, not acting (i.e. acting trivially) on M^{β}_{in}, commuting with $\zeta^{\text{in}}_{i,\text{pad}}(M^{\beta})$, and satisfying: $\mathcal{V}^{\dagger}_{i}[g]\circ\bar{\zeta}^{\text{out}}_{i,\text{pad}}(M^{\beta})$ doesn't act on M^{β}_{out}. We fix an orthonormal (with respect to the Hilbert-Schmidt inner product) basis $(E_m)_{1\leq m\leq\texttt{dim}(M^{\beta}_{\text{out}})^2}$ of $\text{Lin}[\mathcal{H}_{M^{\beta}_{\text{out}}}]$, with $E_0=\mathbb{1}$, and decompose $\mathcal{V}^{\dagger}_{i}[g]$ as*

$$\mathcal{V}^{\dagger}_{i}[g] = \sum_m \chi_m\otimes E_m\,, \tag{90}$$

with the χ_m's acting on $\mathcal{H}_P\otimes\left(\bigotimes_{O^{\gamma}>B(i+1)}\mathcal{H}_{O^{\gamma}_{\text{out}}}\right)$. With padding, we can also write $\mathcal{V}^{\dagger}_{i}[g]=\sum_m\chi_{m,\text{pad}}\circ E_{m,\text{pad}}$, with the terms commuting. We then have

$$\mathcal{V}_{i+1}^{\dagger}[g] = \chi_{0,\mathrm{pad}} + \sum_{m\neq 0} E'_{m,\mathrm{pad}} \circ \chi_{m,\mathrm{pad}}\,, \tag{91}$$

where the E'_m's are defined, through the use of (H6) at step i, by $\mathcal{V}_i^{\dagger}[(U^{\beta\dagger}E_m U^{\beta})_{\mathrm{pad}}] \circ \zeta_{i,\mathrm{pad}}^{\mathrm{out}}(M^{\beta}) = E'_{m,\mathrm{pad}} \circ \zeta_{i,\mathrm{pad}}^{\mathrm{out}}(M^{\beta})$, with the E'_m's only acting on $\mathcal{H}_P$ (because $\mathcal{P}_i^{\mathrm{str}}(M^{\beta}) = \emptyset$).

Proof. We will compute $\mathcal{V}_{i+1}^{\dagger}[g]$ by looking at how g 'moves through' $\mathcal{S}_{\mathrm{pad}}[(V_{N,i+1})_N]$. First, we rewrite (89) more compactly as

$$\begin{aligned}
&\mathcal{S}_{\mathrm{pad}}[(V_{N,i+1})_N] \\
&= \mathrm{Tr}_{M_{\mathrm{out}}^{\beta}}\left[\left(\zeta_{i,\mathrm{pad}}^{\mathrm{in}}(M^{\beta}) \otimes U_{\mathrm{pad}}^{\beta} + \bar{\zeta}_{i,\mathrm{pad}}^{\mathrm{in}}(M^{\beta}) \otimes \frac{1}{\mathtt{dim}(M^{\beta})}\Theta_{M^{\beta},\mathrm{pad}}^{\dagger}\right) \circ \mathcal{S}_{\mathrm{pad}}[(V_{N,i})_N]\right].
\end{aligned} \tag{92}$$

Thus (because g doesn't act on M_{in}^{β}, and commutes with $\zeta_{i,\mathrm{pad}}^{\mathrm{in}}(M^{\beta})$),

$$\begin{aligned}
&g \circ \mathcal{S}_{\mathrm{pad}}[(V_{N,i+1})_N] \\
&= g \circ \mathrm{Tr}_{M_{\mathrm{out}}^{\beta}}\left[\left(\zeta_{i,\mathrm{pad}}^{\mathrm{in}}(M^{\beta}) \circ U_{\mathrm{pad}}^{\beta} + \bar{\zeta}_{i,\mathrm{pad}}^{\mathrm{in}}(M^{\beta}) \circ \frac{1}{\mathtt{dim}(M^{\beta})}\Theta_{M^{\beta},\mathrm{pad}}^{\dagger}\right) \circ \mathcal{S}_{\mathrm{pad}}[(V_{N,i})_N]\right] \\
&= \mathrm{Tr}_{M_{\mathrm{out}}^{\beta}}\left[\left(\zeta_{i,\mathrm{pad}}^{\mathrm{in}}(M^{\beta}) \circ U_{\mathrm{pad}}^{\beta} + \bar{\zeta}_{i,\mathrm{pad}}^{\mathrm{in}}(M^{\beta}) \circ \frac{1}{\mathtt{dim}(M^{\beta})}\Theta_{M^{\beta},\mathrm{pad}}^{\dagger}\right) \circ g \circ \mathcal{S}_{\mathrm{pad}}[(V_{N,i})_N]\right] \\
&= \mathrm{Tr}_{M_{\mathrm{out}}^{\beta}}\left[\left(\zeta_{i,\mathrm{pad}}^{\mathrm{in}}(M^{\beta}) \circ U_{\mathrm{pad}}^{\beta} + \bar{\zeta}_{i,\mathrm{pad}}^{\mathrm{in}}(M^{\beta}) \circ \frac{1}{\mathtt{dim}(M^{\beta})}\Theta_{M^{\beta},\mathrm{pad}}^{\dagger}\right) \circ \mathcal{S}_{\mathrm{pad}}[(V_{N,i})_N] \circ \mathcal{V}_i^{\dagger}[g]\right].
\end{aligned} \tag{93}$$

We now consider the decomposition (90) of $\mathcal{V}_i^{\dagger}[g]$, and we look at

$$\begin{aligned}
\sum_{m\neq 0}(\chi_{m,\mathrm{pad}} \circ \bar{\zeta}_{i,\mathrm{pad}}^{\mathrm{out}}(M^{\beta})) \otimes E_m &= \sum_{m\neq 0}(\chi_m \otimes E_m)_{\mathrm{pad}} \circ \bar{\zeta}_{i,\mathrm{pad}}^{\mathrm{out}}(M^{\beta}) \\
&= \mathcal{V}_i^{\dagger}[g] \circ \bar{\zeta}_{i,\mathrm{pad}}^{\mathrm{out}}(M^{\beta}) - (\chi_0 \otimes \mathbb{1}_{M_{\mathrm{out}}^{\beta}})_{\mathrm{pad}} \circ \bar{\zeta}_{i,\mathrm{pad}}^{\mathrm{out}}(M^{\beta})\,.
\end{aligned} \tag{94}$$

Both terms of the second line's RHS act trivially on M_{out}^{β}: the first term by assumption, and the second because it is a composition of operators acting trivially on M_{out}^{β}. From the form of the LHS, we can thus deduce: $\forall m \neq 0, \chi_{m,\mathrm{pad}} \circ \bar{\zeta}_{i,\mathrm{pad}}^{\mathrm{out}}(M^{\beta}) = 0$, which can be rewritten as

$$\forall m \neq 0, \chi_{m,\mathrm{pad}} = \chi_{m,\mathrm{pad}} \circ \zeta_{i,\mathrm{pad}}^{\mathrm{out}}(M^{\beta})\,. \tag{95}$$

In the same way we can prove that $\chi_{m,\mathrm{pad}} = \zeta_{i,\mathrm{pad}}^{\mathrm{out}}(M^{\beta}) \circ \chi_{m,\mathrm{pad}}$. We are now in a position to continue the computation started in (93); we write $\kappa_i^{\mathrm{in}}(M^{\beta}) := \zeta_{i,\mathrm{pad}}^{\mathrm{in}}(M^{\beta}) \otimes U_{\mathrm{pad}}^{\beta} + \bar{\zeta}_{i,\mathrm{pad}}^{\mathrm{in}}(M^{\beta}) \otimes \frac{1}{\mathtt{dim}(M^{\beta})}\Theta_{M^{\beta},\mathrm{pad}}^{\dagger}$.

$$
\begin{aligned}
& g \circ \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N] \\
& \overset{(95)}{=} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\left(\zeta^{\text{in}}_{i,\text{pad}}(M^\beta) \circ U^\beta_{\text{pad}} + \bar{\zeta}^{\text{in}}_{i,\text{pad}}(M^\beta) \circ \frac{1}{\mathtt{dim}(M^\beta)} \Theta^\dagger_{M^\beta,\text{pad}} \right) \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N] \circ \chi_{0,\text{pad}} \right] \\
& + \sum_{m \neq 0} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\zeta^{\text{in}}_{i,\text{pad}}(M^\beta) \circ U^\beta_{\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N] \circ \chi_{m,\text{pad}} \circ E_{m,\text{pad}} \right] \\
& = \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N] \circ \chi_{0,\text{pad}} \\
& + \sum_{m \neq 0} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[E_{m,\text{pad}} \circ \zeta^{\text{in}}_{i,\text{pad}}(M^\beta) \circ U^\beta_{\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N] \circ \chi_{m,\text{pad}} \right] \\
& = [\ldots] + \sum_{m \neq 0} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\zeta^{\text{in}}_{i,\text{pad}}(M^\beta) \circ U^\beta_{\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N] \circ \mathcal{V}^\dagger_i[(U^{\beta\dagger} E_m U^\beta)_{\text{pad}}] \right] \circ \chi_{m,\text{pad}} \\
& = [\ldots] + \sum_{m \neq 0} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\kappa^{\text{in}}_{i,\text{pad}}(M^\beta) \circ \zeta^{\text{in}}_{i,\text{pad}}(M^\beta) \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N] \circ \mathcal{V}^\dagger_i[(U^{\beta\dagger} E_m U^\beta)_{\text{pad}}] \right] \circ \chi_{m,\text{pad}} \\
& = [\ldots] + \sum_{m \neq 0} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\kappa^{\text{in}}_{i,\text{pad}}(M^\beta) \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N] \circ \mathcal{V}^\dagger_i[(U^{\beta\dagger} E_m U^\beta)_{\text{pad}}] \circ \zeta^{\text{out}}_{i,\text{pad}}(M^\beta) \right] \circ \chi_{m,\text{pad}} \\
& \overset{(\text{H6})}{=} [\ldots] + \sum_{m \neq 0} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\kappa^{\text{in}}_{i,\text{pad}}(M^\beta) \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N] \circ E'_{m,\text{pad}} \circ \zeta^{\text{out}}_{i,\text{pad}}(M^\beta) \right] \circ \chi_{m,\text{pad}} \\
& = [\ldots] + \sum_{m \neq 0} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\kappa^{\text{in}}_{i,\text{pad}}(M^\beta) \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N] \right] \circ E'_{m,\text{pad}} \circ \zeta^{\text{out}}_{i,\text{pad}}(M^\beta) \circ \chi_{m,\text{pad}} \\
& \overset{(92)}{=} [\ldots] + \sum_{m \neq 0} \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N] \circ E'_{m,\text{pad}} \circ \zeta^{\text{out}}_{i,\text{pad}}(M^\beta) \circ \chi_{m,\text{pad}} \\
& \overset{(95)}{=} [\ldots] + \sum_{m \neq 0} \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N] \circ E'_{m,\text{pad}} \circ \chi_{m,\text{pad}} \\
& = \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N] \circ \left(\chi_{0,\text{pad}} + \sum_{m \neq 0} E'_{m,\text{pad}} \circ \chi_{m,\text{pad}} \right) .
\end{aligned}
\tag{96}
$$

In the previous computation, we used (H6) to replace $\mathcal{V}^\dagger_i[(U^{\beta\dagger} E_m U^\beta)_{\text{pad}}] \circ \zeta^{\text{out}}_{i,\text{pad}}(M^\beta)$ with $E'_{m,\text{pad}} \circ \zeta^{\text{out}}_{i,\text{pad}}(M^\beta)$, with the E'_m's only acting on $\mathcal{H}_P$ (because $\mathcal{P}^{\text{str}}_i(M^\beta) = \emptyset$). This then allowed to get the term out of the trace. The computation allows us to conclude that (91) holds.

□

H2 We now turn to (H2). We will take: $\forall N^\alpha > B(i+1), \zeta^{\text{in}}_{i+1}(N^\alpha) := \zeta^{\text{in}}_i(N^\alpha)$. There are two things to check in order to ensure that this makes sense. The first is that the $\zeta^{\text{in}}_i(N^\alpha)$ are indeed all defined, which holds here because i is not a special step. The second thing to check is that for an arbitrary N^α, $\zeta^{\text{in}}_i(N^\alpha)$ wasn't acting on M^β_{in}. This follows from the fact that $i+1$ is not a special step. Indeed, the way we defined the ordering of the branches ensures that $M^\beta \notin \mathcal{F}^{\text{str}}_i(N^\alpha)$. This ensures that $\zeta^{\text{in}}_i(N^\alpha)$ doesn't act on M^β_{in} if N^α is in a green layer; while if N^α is in a red layer, then the fact that $i+1$ is not a special step implies that M^β is not in this red layer, i.e. that $M^\beta \notin \mathcal{F}_i(N^\alpha)$ and thus that $\zeta^{\text{in}}_i(N^\alpha)$ doesn't act on M^β_{in}.

We then want to define, from there, $\zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha) := \mathcal{V}^\dagger_{i+1}[\zeta^{\text{in}}_{i+1,\text{pad}}(N^\alpha)]$, $\forall N^\alpha$. The fact (which derives from (H1)) that $\mathcal{V}_{i+1}$ is an isomorphism of operator algebras will then

ensure that the $\zeta^{\text{out}}_{i+1,\text{pad}}$'s are pairwise commuting orthogonal projectors, as the $\zeta^{\text{in}}_{i+1,\text{pad}}$'s are. What is left to prove is that, fixing an N^α whose layer is green, $\mathcal{V}^\dagger_{i+1}[\zeta^{\text{in}}_{i+1,\text{pad}}(N^\alpha)]$ (which formally acts on the whole $\mathcal{H}^{\text{out}}_{i+1}$) can indeed be seen as the padding of an operator acting on $\mathcal{H}_P \otimes \left(\bigotimes_{O^\gamma \in \mathcal{P}_{i+1}(N^\alpha)} \mathcal{H}_{O^\gamma_{\text{out}}}\right)$ – i.e., that it acts trivially on other factors; and similarly, that for an N^α whose layer is red, $\mathcal{V}^\dagger_{i+1}[\zeta^{\text{in}}_{i+1,\text{pad}}(N^\alpha)]$ can be seen as only acting on $\mathcal{H}_P \otimes \left(\bigotimes_{O^\gamma \in \mathcal{P}^{\text{str}}_{i+1}(N^\alpha)} \mathcal{H}_{O^\gamma_{\text{out}}}\right)$.

For this, fixing an $N^\alpha > B(i+1)$, we will make use of Lemma B.5 to compute $\mathcal{V}^\dagger_{i+1}[\zeta^{\text{in}}_{i,\text{pad}}(N^\alpha)]$. The latter satisfies the lemma's assumptions: $\zeta^{\text{in}}_{i,\text{pad}}(N^\alpha)$ doesn't act on M^β_{in} and commutes with $\zeta^{\text{in}}_{i,\text{pad}}(M^\beta)$ by (H1) at step i, and $\mathcal{V}^\dagger_i[\zeta^{\text{in}}_{i,\text{pad}}(N^\alpha)] = \zeta^{\text{out}}_{i,\text{pad}}(N^\alpha)$, by (H3) at step i, satisfies: $\zeta^{\text{out}}_{i,\text{pad}}(N^\alpha) \circ \bar{\zeta}^{\text{out}}_{i,\text{pad}}(M^\beta)$ acts trivially on M^β_{out}.

By Lemma B.5, writing $\zeta^{\text{out}}_i(N^\alpha) = \sum_m \chi_m \otimes E_m$, we can thus conclude

$$\mathcal{V}^\dagger_{i+1}[\zeta^{\text{in}}_{i+1,\text{pad}}(N^\alpha)] = \chi_{0,\text{pad}} + \sum_{m\neq 0} E'_{m,\text{pad}} \circ \chi_{m,\text{pad}} \,. \tag{97}$$

If N^α is in a green layer, then in this expression, the χ_m's act on $\mathcal{H}_P \otimes \left(\bigotimes_{O^\gamma \in \mathcal{P}_{i+1}(N^\alpha)} \mathcal{H}_{O^\gamma_{\text{out}}}\right)$ and the E'_m's act on $\mathcal{H}_P$; thus, $\mathcal{V}^\dagger_{i+1}[\zeta^{\text{in}}_{i+1,\text{pad}}(N^\alpha)]$ only acts non-trivially on $\mathcal{H}_P \otimes \left(\bigotimes_{O^\gamma \in \mathcal{P}_{i+1}(N^\alpha)} \mathcal{H}_{O^\gamma_{\text{out}}}\right)$. If N^α is in a red layer, the same can be said replacing $\mathcal{P}$'s with $\mathcal{P}^{\text{str}}$'s. This concludes the proof of (H2).

H3 The proof of (H3) is direct for the ζ^{in}_{i+1}'s, as they are equal to the ζ^{in}_i's. For the ζ^{out}_{i+1}'s, fixing N^α and O^γ, one can compute $\zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha) \circ \bar{\zeta}^{\text{out}}_{i+1,\text{pad}}(O^\gamma) = \mathcal{V}^\dagger_{i+1}[\zeta^{\text{in}}_{i,\text{pad}}(N^\alpha) \circ \bar{\zeta}^{\text{in}}_{i,\text{pad}}(O^\gamma)]$ by once again invoking Lemma B.5, writing

$$\zeta^{\text{out}}_i(N^\alpha) \circ \bar{\zeta}^{\text{out}}_i(M^\beta) = \sum_m \xi_m \otimes E_m \,, \tag{98}$$

where the ξ_m's act trivially on O^γ_{out} because $\zeta^{\text{out}}_i(N^\alpha) \circ \bar{\zeta}^{\text{out}}_i(M^\beta)$ does, by (H3) at step i. $\zeta^{\text{in}}_{i,\text{pad}}(N^\alpha) \circ \bar{\zeta}^{\text{in}}_{i,\text{pad}}(O^\gamma)$ commutes with $\zeta^{\text{in}}_{i,\text{pad}}(M^\beta)$ and doesn't act on M^β_{in}; to apply the Lemma, we thus have to prove that $\zeta^{\text{out}}_{i,\text{pad}}(N^\alpha) \circ \bar{\zeta}^{\text{out}}_{i,\text{pad}}(O^\gamma) \circ \bar{\zeta}^{\text{out}}_{i,\text{pad}}(M^\beta)$ acts trivially on M^β_{out}. This follows from the rewriting

$$\begin{aligned} &\zeta^{\text{out}}_{i,\text{pad}}(N^\alpha) \circ \bar{\zeta}^{\text{out}}_{i,\text{pad}}(O^\gamma) \circ \bar{\zeta}^{\text{out}}_{i,\text{pad}}(M^\beta) \\ &= \left(\zeta^{\text{in}}_{i,\text{pad}}(N^\alpha) \circ \bar{\zeta}^{\text{in}}_{i,\text{pad}}(M^\beta)\right) \circ \left(\bar{\zeta}^{\text{out}}_{i,\text{pad}}(M^\beta) - \zeta^{\text{out}}_{i,\text{pad}}(O^\gamma) \circ \bar{\zeta}^{\text{out}}_{i,\text{pad}}(M^\beta)\right) \,, \end{aligned} \tag{99}$$

all of the terms in which, one can conclude by (H3) at step i, act trivially on M^β_{out}. Lemma B.5 thus leads to

$$\zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha) \circ \bar{\zeta}^{\text{out}}_{i+1,\text{pad}}(O^\gamma) = \mathcal{V}^\dagger_{i+1}[\zeta^{\text{in}}_{i,\text{pad}}(N^\alpha) \circ \bar{\zeta}^{\text{in}}_{i,\text{pad}}(O^\gamma)] = \xi_{0,\text{pad}} + \sum_{m\neq 0} E'_{m,\text{pad}} \circ \xi_{m,\text{pad}} \,, \tag{100}$$

with neither the ξ_m's nor the E'_m's acting on O^γ_{out}, which concludes the proof of (H3).

H4 The proof of (H4) at step $i+1$ is immediate as it derives from (H4) at step i for the ζ^{in}_{i+1}'s, which are equal to the ζ^{in}_i's.

H5 For the proof of (H5), we take $Q \subseteq \{B(i') \,|\, i' \geq i+1\}$ a set of branches on different nodes, and $\tilde{Q} \subseteq \texttt{Nodes}_\Gamma$ and a function α such that $Q = \{N^{\alpha(N)} \,|\, N \in \tilde{Q}\}$. We will prove the version of (H5) written with ζ^{in}_{i+1}'s. We first consider the case $M \in \tilde{Q}$. Then by (H4) we have $\zeta^{\text{in}}_{i,\text{pad}}(M^{\alpha(M)}) = \zeta^{\text{in}}_{i,\text{pad}}(M^{\alpha(M)}) \circ \bar{\zeta}^{\text{in}}_{i,\text{pad}}(M^\beta)$, and we can therefore write

$$
\begin{aligned}
&\prod_{N\in\tilde{Q}} \zeta^{\text{in}}_{i+1,\text{pad}}(N^{\alpha(N)}) \circ \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N] \\
&= \prod_{N\in\tilde{Q}} \zeta^{\text{in}}_{i,\text{pad}}(N^{\alpha(N)}) \circ \zeta^{\text{in}}_{i,\text{pad}}(M^\beta) \circ \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N] \\
&= \prod_{N\in\tilde{Q}} \zeta^{\text{in}}_{i,\text{pad}}(N^{\alpha(N)}) \circ \zeta^{\text{in}}_{i,\text{pad}}(M^\beta) \circ \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N] \\
&= \prod_{N\in\tilde{Q}} \zeta^{\text{in}}_{i,\text{pad}}(N^{\alpha(N)}) \circ \zeta^{\text{in}}_{i,\text{pad}}(M^\beta) \circ \frac{1}{\texttt{dim}(M^\beta))} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\Theta^\dagger_{M^\beta,\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]\right] \\
&= \prod_{N\in\tilde{Q}} \zeta^{\text{in}}_{i,\text{pad}}(N^{\alpha(N)}) \circ \frac{1}{\texttt{dim}(M^\beta))} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\Theta^\dagger_{M^\beta,\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]\right] \\
&\overset{\text{(H5)}}{=} \frac{1}{\texttt{dim}(M^\beta))} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\Theta^\dagger_{M^\beta,\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_{N\notin\tilde{Q}} \times (V^{\alpha(N)}_{N,i})_{N\in\tilde{Q}}]\right] \\
&= \frac{1}{\texttt{dim}(M^\beta))} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\Theta^\dagger_{M^\beta,\text{pad}} \circ \Theta_{M^\beta,\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i+1})_{N\notin\tilde{Q}} \times (V^{\alpha(N)}_{N,i+1})_{N\in\tilde{Q}}]\right] \\
&= \mathcal{S}_{\text{pad}}[(V_{N,i+1})_{N\notin\tilde{Q}} \times (V^{\alpha(N)}_{N,i+1})_{N\in\tilde{Q}}] \,.
\end{aligned}
\tag{101}
$$

In the case $M \notin \tilde{Q}$, then defining $\alpha(M) := \beta$, we have

$$
\begin{aligned}
&\prod_{N\in\tilde{Q}} \zeta^{\text{in}}_{i+1,\text{pad}}(N^{\alpha(N)}) \circ \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N] \\
&= \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\prod_{N\in\tilde{Q}\cup\{M\}} \zeta^{\text{in}}_{i,\text{pad}}(N^{\alpha(N)}) \circ U^\beta_{\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]\right] \\
&+ \frac{1}{\texttt{dim}(M^\beta)} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\prod_{N\in\tilde{Q}} \zeta^{\text{in}}_{i,\text{pad}}(N^{\alpha(N)}) \circ \bar{\zeta}^{\text{in}}_{i,\text{pad}}(M^\beta) \circ \Theta^\dagger_{M^\beta,\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]\right] \\
&\overset{\text{(H5)}}{=} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[U^\beta_{\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_{N\notin\tilde{Q}\cup\{M\}} \times (V^{\alpha(N)}_{N,i})_{N\in\tilde{Q}\cup\{M\}}]\right] \\
&+ \frac{1}{\texttt{dim}(M^\beta)} \operatorname{Tr}_{M^\beta_{\text{out}}} \left[\Theta^\dagger_{M^\beta,\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_{N\notin\tilde{Q}\cup\{M\}} \times (V^{\alpha(N)}_{N,i})_{N\in\tilde{Q}} \times (\bar{V}^\beta_{M,i})]\right] \\
&= \mathcal{S}_{\text{pad}}[(V_{N,i+1})_{N\notin\tilde{Q}\cup\{M\}} \times (V^{\alpha(N)}_{N,i+1})_{N\in\tilde{Q}} \times (V^\beta_{M,i})] \\
&+ \mathcal{S}_{\text{pad}}[(V_{N,i+1})_{N\notin\tilde{Q}\cup\{M\}} \times (V^{\alpha(N)}_{N,i+1})_{N\in\tilde{Q}} \times (\bar{V}^\beta_{M,i})] \\
&= \mathcal{S}_{\text{pad}}[(V_{N,i+1})_{N\notin\tilde{Q}} \times (V^{\alpha(N)}_{N,i+1})_{N\in\tilde{Q}}] \,.
\end{aligned}
\tag{102}
$$

H6 To prove (H6), we fix a branch $N^\alpha > B(i+1)$ in a green layer, and $f \in \text{Lin}[\mathcal{H}_{N^\alpha_{\text{in}}}]$. We first consider the case $N^\alpha \notin \mathcal{F}^{\text{str}}_i(M^\beta)$. $\zeta^{\text{in}}_i(M^\beta)$ then doesn't act on N^α_{in}: indeed, either M^β is in a green layer and $\zeta^{\text{in}}_i(M^\beta)$ doesn't act outside of $\mathcal{F}^{\text{str}}_i(M^\beta)$, or M^β is in a red layer and then, because $i+1$ is not a special step, N^α is not in this layer and thus $N^\alpha \notin \mathcal{F}_i(M^\beta)$. Furthermore, as we saw that $\mathcal{S}_{\text{pad}}[(V_{N,i})_N)] \circ \bar{\zeta}^{\text{out}}_{i,\text{pad}}(M^\beta)$ was of the form

$W \otimes \Theta_{M^\beta}$, and f doesn't act on M^β_{in}, it follows that $\mathcal{V}_i^\dagger[f] \circ \bar{\zeta}^{\text{out}}_{i,\text{pad}}(M^\beta)$ doesn't act on M^β_{out}. We can therefore apply Lemma B.5 and get

$$\mathcal{V}_{i+1}^\dagger[f_{\text{pad}}] = \phi_{0,\text{pad}} + \sum_{m\neq 0} E'_{m,\text{pad}} \circ \phi_{m,\text{pad}} \overset{\text{(H6)}}{=} \sum_m \mathcal{V}_i^\dagger[(U^{\beta\dagger} E_m U^\beta)_{\text{pad}}] \circ \phi_{m,\text{pad}}\,, \tag{103}$$

where $\mathcal{V}_i^\dagger[f_{\text{pad}}] = \sum_m \phi_m \otimes E_m$. Furthermore, as we've seen, we have $\zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha) = \sum_n \mathcal{V}_i^\dagger[(U^{\beta\dagger} E_n U^\beta)_{\text{pad}}] \circ \chi_n$. We are therefore led to

$$\begin{aligned} \mathcal{V}_{i+1}^\dagger[f_{\text{pad}}] \circ \zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha) &= \sum_{mn} \phi_m \circ \chi_n \circ \mathcal{V}_i^\dagger[(U^{\beta\dagger} \circ E_m \circ E_n \circ U^\beta)_{\text{pad}}] \\ &= \left(\sum_{mn} \sigma_{lmn} \phi_m \circ \chi_n\right) \circ \sum_l \mathcal{V}_i^\dagger[(U^{\beta\dagger} \circ E_l \circ U^\beta)_{\text{pad}}]\,, \end{aligned} \tag{104}$$

where the σ_{lmn}'s are the structure constants on $\text{Lin}[M^\beta_{\text{out}}]$, i.e. $E_m \circ E_n = \sum_l \sigma_{lmn} E_l$. Yet (H6) at step i gives us that there exists f' acting on P and $\mathcal{P}_i^{\text{str}}(N^\alpha)$ (and therefore not on M^β_{out}) such that $\mathcal{V}_i^\dagger[f_{\text{pad}}] \circ \zeta^{\text{out}}_{i,\text{pad}}(N^\alpha) = {f'}_{\text{pad}} \circ \zeta^{\text{out}}_{i,\text{pad}}(N^\alpha)$, which can be rewritten as

$$\sum_l \left(\sum_{mn} \sigma_{lmn} \phi_m \circ \chi_n\right) \otimes E_l = \sum_l \left(f'_{\text{pad}} \circ \chi_l\right) \otimes E_l\,, \tag{105}$$

leading to

$$\forall l, \sum_{mn} \sigma_{lmn} \phi_m \circ \chi_n = f'_{\text{pad}} \circ \chi_l\,. \tag{106}$$

Reinserting this into (104), we find

$$\begin{aligned} \mathcal{V}_{i+1}^\dagger[f_{\text{pad}}] \circ \zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha) &= f'_{\text{pad}} \circ \sum_l \chi_l \circ \mathcal{V}_i^\dagger[(U^{\beta\dagger} \circ E_l \circ U^\beta)_{\text{pad}}] \\ &= f'_{\text{pad}} \circ \zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha)\,, \end{aligned} \tag{107}$$

where f' acts on P and on $\mathcal{P}_i^{\text{str}}(N^\alpha)$, and the latter is equal to $\mathcal{P}_{i+1}^{\text{str}}(N^\alpha)$ as $M^\beta \notin \mathcal{P}_i^{\text{str}}(N^\alpha)$.

We now consider the case $N^\alpha \in \mathcal{F}_i^{\text{str}}(M^\beta)$. We will use the fact that (58) can be equivalently written as $\zeta^{\text{out}}_{i,\text{pad}}(N^\alpha) \circ \mathcal{V}_i^\dagger[f_{\text{pad}}] = f'_{\text{pad}} \circ \zeta^{\text{out}}_{i,\text{pad}}(N^\alpha)$; indeed, $\zeta^{\text{in}}_{i,\text{pad}}(N^\alpha)$ doesn't act on N^α_{in}, so f_{pad} and $\zeta^{\text{in}}_{i,\text{pad}}(N^\alpha)$ commute, so $\mathcal{V}_i^\dagger[f]$ and $\mathcal{V}_i^\dagger[\zeta^{\text{in}}_{i,\text{pad}}(N^\alpha)] = \zeta^{\text{out}}_{i,\text{pad}}(N^\alpha)$ commute as well. We write $\kappa_i^{\text{out}}(M^\beta) := \zeta^{\text{out}}_{i,\text{pad}}(M^\beta) \otimes U^\beta_{\text{pad}} + \bar{\zeta}^{\text{out}}_{i,\text{pad}}(M^\beta) \otimes \frac{1}{\texttt{dim}(M^\beta)} \Theta^\dagger_{M^\beta,\text{pad}}$.

$$\begin{aligned} &\zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha) \circ \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N]^\dagger \circ f_{\text{pad}} \\ &= \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N]^\dagger \circ \zeta^{\text{in}}_{i,\text{pad}}(N^\alpha) \circ f_{\text{pad}} \\ &= \text{Tr}_{M^\beta_{\text{in}}}\left[\kappa^{\text{out}}_{i,\text{pad}}(M^\beta)^\dagger \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]^\dagger\right] \circ \zeta^{\text{in}}_{i,\text{pad}}(N^\alpha) \circ f_{\text{pad}} \\ &= \text{Tr}_{M^\beta_{\text{in}}}\left[\kappa^{\text{out}}_{i,\text{pad}}(M^\beta)^\dagger \circ \zeta^{\text{out}}_{i,\text{pad}}(N^\alpha) \circ \mathcal{V}_i^\dagger[f_{\text{pad}}] \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]^\dagger\right] \\ &\overset{\text{(H6)}}{=} \text{Tr}_{M^\beta_{\text{in}}}\left[\kappa^{\text{out}}_{i,\text{pad}}(M^\beta)^\dagger \circ f'_{\text{pad}} \circ \zeta^{\text{out}}_{i,\text{pad}}(N^\alpha) \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]^\dagger\right] \\ &= \text{Tr}_{M^\beta_{\text{in}}}\left[\kappa^{\text{out}}_{i,\text{pad}}(M^\beta)^\dagger \circ f'_{\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]^\dagger\right] \circ \zeta^{\text{in}}_{i+1,\text{pad}}(N^\alpha)\,, \end{aligned} \tag{108}$$

from which we get

$$
\begin{aligned}
&\zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha) \circ \mathcal{V}^\dagger_{i+1}[f_{\text{pad}}] \\
&= \zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha) \circ \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N]^\dagger \circ f_{\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N] \\
&= \operatorname{Tr}_{M^\beta_{\text{in}}}\left[\kappa^{\text{out}}_{i,\text{pad}}(M^\beta)^\dagger \circ f'_{\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]^\dagger\right] \\
&\circ \operatorname{Tr}_{M^\beta_{\text{in}}}\left[\mathcal{S}_{\text{pad}}[(V_{N,i})_N] \circ \kappa^{\text{out}}_{i,\text{pad}}(M^\beta)\right] \circ \zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha)\,.
\end{aligned}
\tag{109}
$$

We will rewrite the traces in another way, defining $|\Phi^+(M^\beta)\rangle := \sum_k |k\rangle_{M^\beta_{\text{in}}} \otimes |k\rangle_{M^{\beta\prime}_{\text{in}}}$, where $M^{\beta\prime}_{\text{in}} \cong M^\beta_{\text{in}}$, and $(|k\rangle)_k$ is an arbitrary orthonormal basis. The above can then be expressed as

$$
\begin{aligned}
&\zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha) \circ \mathcal{V}^\dagger_{i+1}[f_{\text{pad}}] \\
&= \langle\Phi^+(M^\beta)|_{\text{pad}} \circ \kappa^{\text{out}}_{i,\text{pad}}(M^\beta)^\dagger \circ f'_{\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]^\dagger \circ |\Phi^+(M^\beta)\rangle_{\text{pad}} \\
&\circ \langle\Phi^+(M^\beta)|_{\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N] \circ \kappa^{\text{out}}_{i,\text{pad}}(M^\beta) \circ |\Phi^+(M^\beta)\rangle_{\text{pad}} \circ \zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha) \\
&= \langle\Phi^+(M^\beta)|_{\text{pad}} \circ \kappa^{\text{out}}_{i,\text{pad}}(M^\beta)^\dagger \circ f'_{\text{pad}} \circ \mathcal{V}^\dagger_{i,\text{pad}}\left[|\Phi^+(M^\beta)\rangle_{\text{pad}} \langle\Phi^+(M^\beta)|_{\text{pad}}\right] \\
&\circ \kappa^{\text{out}}_{i,\text{pad}}(M^\beta) \circ |\Phi^+(M^\beta)\rangle_{\text{pad}} \circ \zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha)\,.
\end{aligned}
\tag{110}
$$

In this expression, $\langle\Phi^+(M^\beta)|$ and $|\Phi^+(M^\beta)\rangle$ act on M^β_{out} and $M^{\beta\prime}_{\text{out}}$; $\kappa^{\text{out}}(M^\beta)$ acts on P, M^β_{out} / M^β_{in} (on its domain/codomain), and $\mathcal{P}_i(M^\beta) \subseteq \mathcal{P}^{\text{str}}_i(N^\alpha)$; f' acts on P and $\mathcal{P}^{\text{str}}_i(N^\alpha)$; and $\mathcal{V}^\dagger_{i,\text{pad}}\left[|\Phi^+(M^\beta)\rangle_{\text{pad}} \langle\Phi^+(M^\beta)|_{\text{pad}}\right]$ acts on P, M^β_{out}, $M^{\beta\prime}_{\text{in}}$ and $\mathcal{P}_i(M^\beta) \subseteq \mathcal{P}^{\text{str}}_i(N^\alpha)$. Their composition – which doesn't act on M^β_{out} and $M^{\beta\prime}_{\text{out}}$ as these are explicitly terminated by $\langle\Phi^+(M^\beta)|$ and $|\Phi^+(M^\beta)\rangle$ – thus acts trivially outside of P and $\mathcal{P}^{\text{str}}_i(N^\alpha) \setminus \{M^\beta\} = \mathcal{P}^{\text{str}}_{i+1}(N^\alpha)$. Therefore, we can write

$$
\zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha) \circ \mathcal{V}^\dagger_{i+1}[f_{\text{pad}}] = f''_{\text{pad}} \circ \zeta^{\text{out}}_{i+1,\text{pad}}(N^\alpha)\,,
\tag{111}
$$

with $f'' \in \operatorname{Lin}\left[\mathcal{H}_P \otimes (\bigotimes_{O^\gamma \in \mathcal{P}^{\text{str}}_{i+1}(N^\alpha)} \mathcal{H}_{O^\gamma_{\text{out}}})\right]$.

H7 We take $N^\alpha > M^\beta$ in a red layer, and $f \in \operatorname{Lin}[\mathcal{H}_{N^\alpha_{\text{out}}}]$. Because N^α is in a red layer and $i+1$ is not a special step, we have $N^\alpha \notin \mathcal{P}_i(M^\beta)$. Thus f commutes with $\kappa^{\text{out}}_i(M^\beta)$, as the latter only acts non trivially on P, $\mathcal{P}_i(M^\beta)$ and M^β. Thus,

$$
\begin{aligned}
&f_{\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N]^\dagger \circ \zeta^{\text{in}}_{i+1,\text{pad}}(N^\alpha) \\
&= f_{\text{pad}} \circ \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N]^\dagger \circ \zeta^{\text{in}}_{i,\text{pad}}(N^\alpha) \\
&= \operatorname{Tr}_{M^\beta_{\text{in}}}\left[\kappa^{\text{out}}_{i,\text{pad}}(M^\beta)^\dagger \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]^\dagger \circ \mathcal{V}^\dagger_i[f_{\text{pad}}] \circ \zeta^{\text{in}}_{i,\text{pad}}(N^\alpha)\right] \\
&= \operatorname{Tr}_{M^\beta_{\text{in}}}\left[\kappa^{\text{out}}_{i,\text{pad}}(M^\beta)^\dagger \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]^\dagger \circ \zeta^{\text{in}}_{i,\text{pad}}(N^\alpha) \circ \mathcal{V}^\dagger_i[f_{\text{pad}}]\right] \\
&\overset{\text{(H7)}}{=} \operatorname{Tr}_{M^\beta_{\text{in}}}\left[\kappa^{\text{out}}_{i,\text{pad}}(M^\beta)^\dagger \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]^\dagger \circ \zeta^{\text{in}}_{i,\text{pad}}(N^\alpha) \circ f'_{\text{pad}}\right] \\
&= \operatorname{Tr}_{M^\beta_{\text{in}}}\left[\kappa^{\text{out}}_{i,\text{pad}}(M^\beta)^\dagger \circ \mathcal{S}_{\text{pad}}[(V_{N,i})_N]^\dagger\right] \circ \zeta^{\text{in}}_{i,\text{pad}}(N^\alpha) \circ f'_{\text{pad}} \\
&= \mathcal{S}_{\text{pad}}[(V_{N,i+1})_N]^\dagger \circ \zeta^{\text{in}}_{i,\text{pad}}(N^\alpha) \circ f'_{\text{pad}}\,,
\end{aligned}
\tag{112}
$$

where f' acts on $\mathcal{F}_i^{\rm str}(N^\alpha) = \mathcal{F}_{i+1}^{\rm str}(N^\alpha)$. We can use this to find (noting that $\mathcal{V}_{i+1}[f]$ and $\zeta_{i+1,\rm pad}^{\rm in}(N^\alpha)$ commute because f and $\zeta_{i+1,\rm pad}^{\rm out}(N^\alpha)$ do)

$$\zeta_{i+1,\rm pad}^{\rm in}(N^\alpha) \circ \mathcal{V}_{i+1}[f_{\rm pad}] = \mathcal{V}_{i+1}[f_{\rm pad}] \circ \zeta_{i+1,\rm pad}^{\rm in}(N^\alpha) = \zeta_{i+1,\rm pad}^{\rm in}(N^\alpha) \circ f'_{\rm pad}\,. \tag{113}$$

The special steps The previous proofs were relying on the assumption that neither i nor $i+1$ were special steps. We now consider the other cases. Note first that the proof of (H1) presented earlier is valid in these cases as well.

We start with the case: $i+1$ is a special step (regardless of the status of i). One then only has to define, and check the properties of, the $\zeta_{i+1}(N^\alpha)$'s for $N^\alpha \in \mathcal{L}_{i+1}(M^\beta)$. Note that for such N^α's, the $\zeta_i(N^\alpha)$'s were defined at step i; indeed, either i is not a special step and all the ζ_i's were defined, or i is a special step, which entails that $\mathcal{L}_{i+1}(M^\beta) \subset \mathcal{L}_i(B(i))$, and the ζ_i's were defined for elements of this latter set.

One can then follow a proof strategy that is a time-reversed version of the one presented earlier, except that only the N^α's in $\mathcal{L}_{i+1}(M^\beta)$ are considered. Namely, we will define, for $N^\alpha \in \mathcal{L}_{i+1}(M^\beta)$, $\zeta_{i+1}^{\rm out}(N^\alpha) := \zeta_i^{\rm out}(N^\alpha)$ and $\zeta_{i+1,\rm pad}^{\rm in}(N^\alpha) := \mathcal{V}_i[\zeta_{i+1,\rm pad}^{\rm out}(N^\alpha)]$, and the rest of the proof can be obtained by following the earlier proof, simply replacing in's with out's, looking at daggered versions of maps, etc. Indeed, the induction hypotheses are fully invariant under time-symmetry, except for one crucial thing: the fact that, when looking in the forward direction, all branches of the layers in the strict past of the branch under consideration have been refilled already, and thus have no ζ's. Here, however, we are only redefining, and proving properties of, the $\zeta_{i+1}(N^\alpha)$'s of the layer under consideration; everything thus goes as if the branches in its strict future didn't exist. Moreover, the fact that $i+1$ is a special step implies that $B(i+1)$ is in a red layer, which means that, when considering things from a time-reversed perspective, $B(i+1)$ is in a green layer and thus neither i not $i+1$ are special steps. Thus, for the purposes of defining these $\zeta_{i+1}(N^\alpha)$'s, the situation is exactly symmetric to the one considered previously.

A final case to consider is: i is a special step but $i+1$ is not. In this case, $B(i)$ and $M^\beta = B(i+1)$ are in the same red layer, but $B(i+2)$ and the rest of the $N^\alpha > B(i+1)$ aren't. The interpretation is that we just finished filling up the branches of a red layer, a procedure during which we didn't define ζ's for the branches above it; so that we now have to redefine them. The strategy for this case is to define the $\zeta_{i+1}^{\rm in}(N^\alpha)$'s to be equal, not to the $\zeta_i^{\rm in}(N^\alpha)$'s – which were not defined –, but to the $\zeta_j^{\rm in}(N^\alpha)$, where j is the latest step that was not special, i.e. the latest step at which these were defined. $B(j)$ is then the first branch of the red layer we finished refilling.

One can then follow a strategy similar to the previous proof, now deriving that (H2)-(H7) hold at step $i+1$ from the fact that they hold at step j. Let us highlight the main steps. First, from (89) holding at all steps between j and $i+1$, we can deduce

$$\mathcal{S}_{\rm pad}[(V_{N,i+1})_N] = {\rm Tr}_{B(t)^{\rm in},\, j<t\leq i+1}\left[\mathcal{S}_{\rm pad}[(V_{N,j})_N] \circ \left(\prod_{t=j+1}^{i+1} \kappa_{j,\rm pad}^{\rm out}(B(t))\right)\right]\,, \tag{114}$$

where we also used the fact that, due to how we defined the $\zeta^{\rm out}$'s to remain the same when filling up a red layer, we have $\forall t \in [\![j+1, i+1]\!], \kappa_t^{\rm out}(B(t)) = \kappa_j^{\rm out}(B(t))$.

For (H2), we fix $N^\alpha > B(i+1)$; note that we then have $N^\alpha \in \mathcal{F}_i^{\rm str}(B(i+1))$, because $B(i+1)$ is the last branch in its (red) layer. As before, we have to prove that, defining $\zeta_{i+1,\rm pad}^{\rm in}(N^\alpha) := \zeta_{j,\rm pad}^{\rm in}$ and $\zeta_{i+1,\rm pad}^{\rm out}(N^\alpha) := \mathcal{V}^{i+1}[\zeta_{i+1,\rm pad}^{\rm in}(N^\alpha)]$, the latter doesn't act outside of P and $\mathcal{P}_{i+1}(N^\alpha)$ (or $\mathcal{P}_{i+1}^{\rm str}(N^\alpha)$ if N^α is in a red layer). Using (114) and techniques similar to before, we are led to

$$
\begin{aligned}
\mathcal{V}_{i+1}^{\dagger}[\zeta_{j,\mathrm{pad}}^{\mathrm{in}}(N^\alpha)] = &\left(\prod_{t=j+1}^{i+1} \langle\phi^+(B(t))|_{\mathrm{pad}}\right) \circ \left(\prod_{t=j+1}^{i+1} \kappa_{j,\mathrm{pad}}^{\mathrm{out}}(B(t))^\dagger\right) \\
&\circ \mathcal{V}_{j,\mathrm{pad}}^{\dagger}\left[\prod_{t=j+1}^{i+1} |\Phi^+(B(t))\rangle_{\mathrm{pad}} \langle\Phi^+(B(t))|_{\mathrm{pad}}\right] \circ \zeta_{j,\mathrm{pad}}^{\mathrm{out}}(N^\alpha) \\
&\circ \left(\prod_{t=j+1}^{i+1} \kappa_{j,\mathrm{pad}}^{\mathrm{out}}(B(t))\right) \circ \left(\prod_{t=j+1}^{i+1} |\phi^+(B(t))\rangle_{\mathrm{pad}}\right),
\end{aligned}
\tag{115}
$$

in which $\zeta_{j,\mathrm{pad}}^{\mathrm{out}}(N^\alpha)$ acts only on P and $\mathcal{P}_j(N^\alpha)$ ($\mathcal{P}_j^{\mathrm{str}}(N^\alpha)$ if N^α is in a red layer), and the other terms act only on $\mathcal{L}(B(i+1))$ and on P. Given that all the wires in $\mathcal{L}(B(i+1))$ are explicitly terminated by the ϕ^+'s, it follows that $\mathcal{V}_{i+1}^{\dagger}[\zeta_{j,\mathrm{pad}}^{\mathrm{in}}(N^\alpha)]$ only acts on P and on $\mathcal{P}_j(N^\alpha) \setminus \mathcal{L}(B(i+1)) = \mathcal{P}_{i+1}(N^\alpha)$ (or $\mathcal{P}_j^{\mathrm{str}}(N^\alpha) \setminus \mathcal{L}(B(i+1)) = \mathcal{P}_{i+1}^{\mathrm{str}}(N^\alpha)$ if N^α is in a red layer).

The proof of (H3) is fully analogous: computing $\mathcal{V}_{i+1}^{\dagger}[\zeta_{j,\mathrm{pad}}^{\mathrm{in}}(N^\alpha) \circ \bar{\zeta}_{j,\mathrm{pad}}^{\mathrm{in}}(M^\beta)]$ leads to (115) with $\zeta_{j,\mathrm{pad}}^{\mathrm{out}}(N^\alpha)$ replaced with $\zeta_{j,\mathrm{pad}}^{\mathrm{out}}(N^\alpha) \circ \bar{\zeta}_{j,\mathrm{pad}}^{\mathrm{out}}(M^\beta)$, so that invoking (H3) at step j leads to (H3) at step $i+1$. (H4), as before, is direct, and the proof of (H5) is analogous to the one for the non-special cases.

For the proof of (H6), we take $N^\alpha > B(i+1)$ in a green layer and $f \in \mathrm{Lin}[\mathcal{H}_{N_{\mathrm{in}}^\alpha}]$. Then, the computation is similar to (108), (109) and (110), yielding

$$
\begin{aligned}
&\zeta_{i+1,\mathrm{pad}}^{\mathrm{out}}(N^\alpha) \circ \mathcal{V}_{i+1}^{\dagger}[f_{\mathrm{pad}}] \\
&= \left(\prod_{t=j+1}^{i+1} \langle\phi^+(B(t))|_{\mathrm{pad}}\right) \circ \left(\prod_{t=j+1}^{i+1} \kappa_{j,\mathrm{pad}}^{\mathrm{out}}(B(t))^\dagger\right) \circ \mathcal{V}_{j,\mathrm{pad}}^{\dagger}\left[\prod_{t=j+1}^{i+1} |\Phi^+(B(t))\rangle_{\mathrm{pad}} \langle\Phi^+(B(t))|_{\mathrm{pad}}\right] \\
&\circ f'_{\mathrm{pad}} \circ \left(\prod_{t=j+1}^{i+1} \kappa_{j,\mathrm{pad}}^{\mathrm{out}}(B(t))\right) \circ \left(\prod_{t=j+1}^{i+1} |\phi^+(B(t))\rangle_{\mathrm{pad}}\right) \circ \zeta_{i+1,\mathrm{pad}}^{\mathrm{out}}(N^\alpha),
\end{aligned}
\tag{116}
$$

where $f' \in \mathrm{Lin}[\mathcal{H}_P \otimes (\bigotimes_{O^\gamma \in \mathcal{P}_j^{\mathrm{str}}(N^\alpha)})]$ was defined through $\zeta_{j,\mathrm{pad}}^{\mathrm{out}}(N^\alpha) \circ \mathcal{V}_j^{\dagger}[f_{\mathrm{pad}}] = f'_{\mathrm{pad}} \circ \zeta_{j,\mathrm{pad}}^{\mathrm{out}}(N^\alpha)$ by using (H6) at step j. Once again, by looking at where the operators are acting, we can conclude that this defines a f'' acting only on P and on $\mathcal{P}_j^{\mathrm{str}}(N^\alpha) \setminus \mathcal{L}(B(i+1)) = \mathcal{P}_{i+1}^{\mathrm{str}}(N^\alpha)$.

Finally, for (H7), one can follow computations (112) and (113), to get

$$
\zeta_{i+1,\mathrm{pad}}^{\mathrm{in}}(N^\alpha) \circ \mathcal{V}_{i+1}[f_{\mathrm{pad}}] = \zeta_{j,\mathrm{pad}}^{\mathrm{in}}(N^\alpha) \circ f'_{\mathrm{pad}}, \tag{117}
$$

where f', obtained by the use of (H7) at step j, only acts on $\mathcal{F}_j^{\mathrm{str}}(N^\alpha) = \mathcal{F}_{i+1}^{\mathrm{str}}(N^\alpha)$. This concludes the proof of the induction step.

Conclusion As the base case and induction step are true, the induction hypotheses are true at every step up to n. In particular, (H1) at step n then reads:

$$
\mathcal{S}[(U_N)_N] \text{ is unitary.} \tag{118}
$$

As this was done for $\mathcal{S} = \mathcal{S}_{\left(\Gamma,(\lambda_N)_N\right)}$ for an arbitrary valid routed graph $(\Gamma, (\lambda_N)_N)$, and for an arbitrary collection of routed unitaries $U_N : \mathcal{H}_N^{\text{in}} \xrightarrow{\lambda_N} \mathcal{H}_N^{\text{out}}$ following the λ_N's, we can invoke Lemma B.2 and conclude that Theorem B.1 holds.

$\square$