Quantum Bayesian Networks: Compositionality and Typing via Linear Logic

Rémi Di Guardia
Université Paris Cité, CNRS, Inria, IRIF, F-75013, Paris, France

Thomas Ehrhard
Université Paris Cité, CNRS, IRIF, F-75013, Paris, France

Claudia Faggian
Université Paris Cité, CNRS, IRIF, F-75013, Paris, France

Abstract

Quantum Bayesian networks [15] provide a mathematical formalism to describe causal relations, to analyse correlations, and to predict the probabilities of measurement outcomes, in systems involving both *classical and quantum* data. They generalize Pearl's Bayesian networks [25]—prominent graphical models for classical probabilistic reasoning and inference.

The goal of this paper is to bring compositional principles and a typing discipline into this setting. A key feature of our compositional semantics is that when all causes are classical, it coincides with the standard factor-based semantics of Bayesian networks, while in the purely quantum case it reduces to tensor networks. We then propose a typed formalism based on linear logic proof-nets, where types ensure well-behaved composition of systems, and which we prove sound and complete with respect to quantum Bayesian networks.

2012 ACM Subject Classification Mathematics of computing → Bayesian networks; Theory of computation → Quantum computation theory; Theory of computation → Linear logic

Keywords and phrases Quantum Bayesian Networks, Quantum Causal Models, Bayesian Networks, Proof-Nets, Linear Logic

Related Version : `https://arxiv.org/abs/2604.26059` [9]

Funding by the Plan France 2030 through the PEPR project EPiQ (ANR-22-PETQ0007) and by the European Union through the MSCA SE project QCOMICAL (Grant Agreement ID: 101182520).

Acknowledgements We are grateful to Benoît Valiron for insightful discussions and to the anonymous referees for their valuable comments and the suggested references.

1 Introduction

Pearl's *Bayesian networks* [24, 25] provide a framework for reasoning under conditions of uncertainty and partial knowledge, with a wide range of applications from statistics to epidemiology, economics and computer science. Bayesian networks have a *dual nature*, serving both as probabilistic graphical models for classical probabilistic reasoning and inference, and as causal models, precising the connections between observed data and causal relations. When reasoning on *quantum systems*, the classical framework is not general enough to account for entanglement and the non-local correlations observed in Bell experiments. The development of quantum causal models (see *e.g.* [2] and references therein) is an active research area across quantum information and the foundations of quantum theory, advancing along various axes, whose motivations span from foundational questions and non-locality, to enabling device-independent cryptographic protocols, to facilitating data-driven discovery.

In this paper, we focus on quantum Bayesian networks, a direct generalization of Pearl's networks introduced in foundational work by Henson, Lal, and Pusey [15]. They provide a mathematical framework to describe causal relations, to analyse correlations, and to

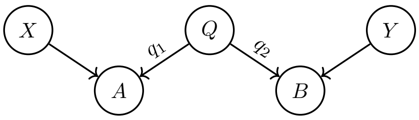


Figure 1 Bell set-up (from [15])

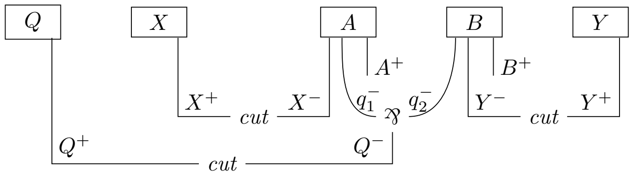


Figure 2 Bell set-up as a quantum proof-net

predict the probabilities of measurement outcomes, in systems involving both classical and quantum data. The formalism builds on previous work by Leifer and Spekkens [22], where the perspective is that of *quantum theory as a theory of inference.* Quantum theory is indeed fundamentally probabilistic at its core, as it is concerned with predicting the probabilities of measurement outcomes on a physical system. In this sense, the prediction task can be framed as a problem of *probabilistic inference over models involving both classical and quantum data*, as Ex. 1 illustrates. Probabilistic inference then offers mathematical and logical tools to comprehend, predict, and control quantum phenomena, essential both to the theoretical understanding and to quantum technologies.

▶ **Example 1** (Alice & Bob: the Bell set-up)**.** The directed acyclic graph in Fig. 1 describes the well-known set-up for the Bell experiment. Alice and Bob—who stand in widely separated laboratories—are each able to perform two possible measurements on a qubit (for example, measuring it with respect to two different bases). Their colleague Quentin prepares a pair of (possibly entangled) qubits, sending one to Alice and the other to Bob. When Alice receives her qubit q_1, she chooses to randomly perform one of the two possible measurements, by flipping a coin X. When Bob receives his qubit q_2, he also performs a measurement, by flipping a coin Y. The result of the experiment is

$$\Pr(a, b \mid x, y) = \Pr(a, b, x, y) / \Pr(x, y) \tag{1}$$

i.e. the probability that the (classical) outcomes of Alice and Bob measurements are respectively a and b, given outcomes x for X and y for Y.

Bayesian Networks. In the theory of Bayesian networks, the causal structure is encoded by a *directed acyclic graph* (DAG) where nodes represent random variables and edges express conditional dependencies. The strength of the dependencies (or the degree of knowledge) is quantified by conditional probability tables. A key benefit of Bayesian networks is to provide a compact representation of large probability distributions, and efficient inference algorithms (both exact and approximate) to answer queries about the underlying distribution without explicitly constructing it in full. Critically, the DAG includes both *observable* variables of interest, and *hidden* (unobserved) ones; in figures, we adopt the common convention of denoting classical hidden variables by shaded nodes. In the quantum setting—as in Bell's theorems—hidden variables play a central role, and they may correspond to quantum systems.

Semantics and Inference. The semantics of a Bayesian network is the probability distribution it defines. More accurately, what one seeks is the *marginal distribution over the variables of interest.* Exact inference computes it precisely— this involves two key operations:

product (*composing*) + summing-out (*hiding*) irrelevant variables.

The formalization and theory of inference rely on a class of functions, known as factors (see Sec. 2.1), which is an abstraction of conditional probability distributions. Tractability and efficiency rely on their properties, and specifically on two key aspects: the product of factors *inherently shares variables*, and product and sum *distribute* under suitable conditions—pushing the sum on smaller components reduces the size of computations.

Quantum Bayesian Networks, issues. Quantum Bayesian networks are still an emerging field, not as developed as their classical counterparts. A crucial missing feature is the ability to compute the semantics of the model (the desired marginal distribution) through intermediate, partial computations, without ever computing the full joint distribution. Put differently, what is lacking is *compositionality*, which would enable computing a model's denotation as a function of its subparts, alongside modular reasoning. A closely related question concerns *modularity*: when can causal descriptions of systems as subparts be used to construct larger models? A well-established tool to ensure modularity are *types*, that specify precise contracts (*e.g.* input/output behaviors) for the components of a system.

The goal of this paper. We have two main objectives in this paper.

- To address the lack of compositionality and modularity in the setting of quantum Bayesian networks by introducing methods and concepts from denotational semantics and proof theory, thereby enabling compositional principles and a typing discipline.
- To develop a framework fully compatible with Bayesian networks and Bayesian inference, thus paving the way for the application of the techniques developed in that context.

Compositionality and modularity facilitate reasoning about complex systems and their properties, ensuring that the meaning of the system can be derived systematically and in a principled way from the meanings of its parts, and allowing components to be analyzed and replaced independently. Compositionality also enables modular reasoning about inference.

Types serve as abstract characterizations of systems behaviors, constraining and guiding their formation. Types discipline statically guarantees semantic properties such as termination, consistency, and compositional correctness. Well-typed programs exhibit well-behaved execution and preserve quantitative (*e.g.* probabilistic) invariants throughout their evaluation.

Encompassing the semantics of Bayesian networks. As seen in Ex. 1 (the Bell set-up), even for systems involving quantum sources of causality, only the classical outcomes of measurements can be observed. Thus, what a model ultimately defines is a probability distribution over classical variables—such as Eq. (1). From the inference perspective, it is desirable to have a framework enabling (when relevant) the extensive set of inference techniques and algorithms developed for Bayesian networks. To this end, our semantics integrates a key notion of inference algorithms, that of *factor*, which we discuss in Sec. 2.1.

Contributions and challenges. Our first contribution is to develop a *compositional semantics*, which allows for the interpretation and modular combination of components. We adapt Selinger's semantics in [26] to take into account the factor-based approach of Bayesian networks. The main technical challenge is to conciliate two very different behaviors:

- *Classical variables share their values, and they do so in an efficient way*: the mathematical setting underlying Bayesian networks integrates this feature in the very definition of factors product, and exploits it to obtain compact representations and efficient calculations.
- *Quantum data cannot be shared*: a defining feature of quantum computing is the *No-Cloning Theorem*, implying qubits cannot be duplicated nor broadcast to multiple receivers.

We satisfy both requirements by introducing *quantum factors* in Sec. 3. The mathematical developments in that section are our main and most technical results. Remarkably, when all causes are classical, our framework exactly coincides with the standard *factor-based* semantics of Bayesian networks, while in the purely quantum case it behaves like tensor networks. In Sec. 4, we rely on quantum factors to redefine quantum Bayesian networks. Our formalism is equivalent to that in [15], however our semantics enables a compositional interpretation by sub-components, unlike the original definition (see [15, page 12]), as we discuss in Sec. 2.3. A crucial aspect to achieve compositionality is that quantum factors are closed under both product and sum-out (*i.e.* marginalization of unobserved variables). Finally, in Sec. 5, we explore a *typed graphical formalism* based on proof-nets of linear logic, where types ensure well-behaved compositions of systems. Our key result is that the formalism is *sound and complete w.r.t.* quantum Bayesian networks: every quantum Bayesian network can be represented as a proof-net, and every closed proof-net corresponds to a quantum Bayesian network.

Motivational examples: compositionality and modularity

Two examples can illustrate the desiderata and issues with compositionality and modularity.

Compositionality. Consider the DAG in Fig. 3. The nodes X and Y produce a classical output, while the nodes Q_1 and Q_2 have a quantum nature. A natural question is: *Can we compute the semantics of the model in terms of sub-components*—as for example the highlighted sub-graphs? The approach in [15] does not adapt well to an interpretation by components because it has a global nature, as we will discuss in Sec. 2.3.

Do parts compose well? When considering components (possibly with open inputs[1]), a natural question somehow dual to the previous one is whether independently defined components compose well. Consider the three DAGs in Fig. 4: $\mathcal{N}_0$ awaits an input A and outputs C, $\mathcal{N}_1$ and $\mathcal{N}_2$ both await an input C and output A and D. The graph obtained by plugging together $\mathcal{N}_0$ and $\mathcal{N}_1$ (matching inputs and outputs) is a DAG, while the graph that plugs together $\mathcal{N}_0$ and $\mathcal{N}_2$ has a directed cycle. By moving to *typed graphs*, we guarantee that composing graphs of compatible types produces a DAG (Sec. 5.2). Rather than defining yet-another-syntax, we encode quantum Bayesian networks into the graph syntax of linear logic: proof-nets. This builds on a recent line of work connecting (classical) Bayesian networks with proof-nets [11, 12, 8]. A proof-net is typed by a sequent in Multiplicative Linear Logic. As we will see (Sec. 5.2 and Fig. 7), the DAGs in Fig. 4 admit the following typing:

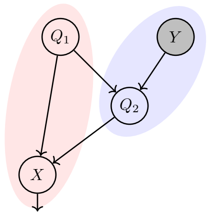
$$\mathcal{N}_0 \vdash A \multimap C \qquad\qquad \mathcal{N}_1 \vdash (A \multimap C) \multimap D \qquad\qquad \mathcal{N}_2 \vdash C \multimap (A \otimes D)$$

The DAGs $\mathcal{N}_0$ and $\mathcal{N}_1$ compose together, producing a DAG of output D, while $\mathcal{N}_2$ cannot be given any type that matches the one of $\mathcal{N}_0$.

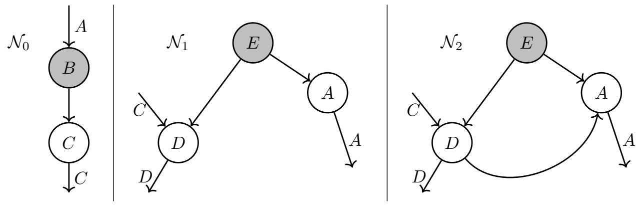
2 Preliminaries

We recall some basics on Bayesian networks and on quantum computation, respectively referring to [7] (or [6] for a compact introduction) and to [23, 30] for further reading. We then present quantum Bayesian networks from [15].

[1] Bayesian networks with open inputs are called *conditional Bayesian networks*, see *e.g.* [19].



■ **Figure 3** Compositionality



■ **Figure 4** Modularity

2.1 Classical data and Bayesian Networks

Bayesian networks (BNs) are probabilistic graphical models. Bayesian models provide a formalism for reasoning under conditions of uncertainty or partial knowledge: given a system under study, how *likely* is it that a particular feature is in a particular state? Every feature of the system is represented by a *random variable*. For the purpose of modeling, each random variable can be seen as a *name* for an atomic proposition (*e.g.* "Rain") which assumes values from a set of states (*e.g.* $\{\mathtt{t}, \mathtt{f}\}$). The full system is modeled as *a joint probability distribution* over the variables of interest; each element in the sample space represents a possible state.

Random Variables (r.v.s). Assuming a countable set of names $X, Y, \ldots$, a *random variable* (r.v.) is the pair of a name X and a finite set of values $\mathsf{Val}(X)$. *W.l.o.g.*, we will assume all random variables to be binary, with $\mathsf{Val}(X) = \{x^{\mathtt{t}}, x^{\mathtt{f}}\}$ for a name X. We then silently identify a name X with the r.v. $(X, \mathsf{Val}(X))$.

We adopt the standard convention of capital letters (*e.g.* X, Y) denoting random variables, and lowercase letters (*e.g.* x, y) for particular *values* of those variables; $\Pr(x)$ stands for $\Pr(X = x)$. A finite set of names $\mathbf{X} = \{X_1, \ldots, X_n\}$ defines a "compound" r.v. where $\mathsf{Val}(\mathbf{X})$ is the Cartesian product $\mathsf{Val}(X_1) \times \cdots \times \mathsf{Val}(X_n)$; we denote by $\mathbf{x}$ a tuple in $\mathsf{Val}(\mathbf{X})$.

Bayesian Networks. A Bayesian network over a set of r.v.s $\mathbf{X}$ is a DAG whose nodes are the r.v.s and whose directed edges represent conditional dependencies. It represents in a compact and factorized way a joint probability distribution $\Pr(\mathbf{X})$, by exploiting conditional (in)dependencies in the distribution. This is achieved by associating with each node a *function*—a conditional probability table—which quantifies the dependencies of each node from its parents.

Formally, a **Bayesian network** $\mathcal{B}$ over the set of r.v.s $\mathbf{X}$ is a pair $(\mathcal{G}, \Phi)$ where:
- $\mathcal{G}$ is a directed acyclic graph (DAG), whose set of nodes is $\mathbf{X}$;
- Φ assigns to each $X \in \mathbf{X}$ a *conditional probability table (CPT)* ϕ^X, for X given its parents.

▶ **Theorem 2** (Semantics of BNs [24]). *A BN $\mathcal{B}$ over the set of r.v.s $\mathbf{X}$ defines a unique probability distribution* $\Pr(\mathbf{X})$ *given by the* product *of the CPTs associated to the nodes:*

$$\Pr(\mathbf{X}) = \prod_{X \in \mathbf{X}} \phi^X \tag{$*$}$$

▶ Remark 3 (Markov condition). Given a Bayesian network over $\mathbf{X}$, each variable X is assumed to be independent of its nondescendants, given its parents. A probability distribution $\Pr(\mathbf{X})$ can be expressed as $(*)$ if and only if it satisfies such a Markov condition.

The *marginal distribution* over the *variables of interest* $\mathbf{Y} \subseteq \mathbf{X}$ is obtained by summing-out the irrelevant variables: $\Pr(\mathbf{Y}) = \sum_{\mathbf{X}-\mathbf{Y}} \Pr(\mathbf{X})$.

From CPTs to Factors. The strength of BNs is to represent large probability distributions compactly, and to allow for inference algorithms that do not construct the full joint distribution explicitly. CPTs alone are insufficient for this purpose, motivating the definition of a class of functions called *factors*, a general mathematical abstraction representing both the initial parameters (the CPTs) and the partial results generated during the inference. Formally, a *factor* ϕ over the r.v.s $\mathbf{X}$ is a function mapping each $\mathbf{x} \in \mathsf{Val}(\mathbf{X})$ to a non-negative real:

$$\phi : \mathsf{Val}(\mathbf{X}) \to \mathbb{R}_0^+ \tag{2}$$

A CPT θ for X given $\mathbf{Y}$ is a factor over $X, \mathbf{Y}$ which satisfies $\sum_x \theta(x, \mathbf{y}) = 1$ for all $\mathbf{y} \in \mathsf{Val}(\mathbf{Y})$.

Unlike CPTs, factors are closed under both *product and sum-out* operations. This is crucial for algorithms (*e.g.* Variable Elimination) that rely on pushing the sum (marginalization) to smaller components to reduce the cost of computations. The so generated partial results are factors, but not CPTs. Moving to factors is also necessary to incorporate evidence.

Sharing variables. We refer to [6] for the formal definitions of sum and product of factors, which are the natural extensions of the corresponding operations for probabilities. Tractability relies on the properties of these operations, and specifically on two key aspects: the product of factors *inherently shares variables*, and product and sum *distribute* under suitable conditions. Distributivity allows for pushing the sum on smaller components, reducing the size of computations. Let us briefly discuss how sharing is built in the definition of product, and what this means. Assume we want to compute the distribution $\Pr(X, Y, Z)$ defined as the product $\Pr(Z)\Pr(X \mid Z)\Pr(Y \mid Z)$. This amounts to computing 2^3 entries:

$$\Pr(x, y, z) = \Pr(z)\Pr(x \mid z)\Pr(y \mid z) \quad \text{for } x \in \mathsf{Val}(X), y \in \mathsf{Val}(Y), z \in \mathsf{Val}(Z)$$

Observe how the values corresponding to the same r.v. Z are shared: we compute $\Pr(z^{\mathtt{t}})\Pr(x \mid z^{\mathtt{t}})\Pr(y \mid z^{\mathtt{t}})$ and $\Pr(z^{\mathtt{f}})\Pr(x \mid z^{\mathtt{f}})\Pr(y \mid z^{\mathtt{f}})$, but never $\Pr(z^{\mathtt{t}})\Pr(x \mid z^{\mathtt{t}})\Pr(y \mid z^{\mathtt{f}})$!

It is important to stress that the definition of the factors product is starkly different with that of the tensor product of matrices, which is the natural product for quantum systems. We compare the two operations in Appendix A—the *computational cost* being also starkly different (as discussed in [12, Example 9.7]).

2.2 Quantum data

An (isolated) quantum system is associated with a complex vector space equipped with an inner product $\langle\cdot|\cdot\rangle$, *i.e.* a *Hilbert space* $\mathcal{H}$ called the **state space** of the system. The simplest quantum system is the *qubit*, which has a two-dimensional state space corresponding to $\mathbb{C}^2$.

The **states** of a quantum system can be described in two mathematically equivalent ways, either as *vectors in* the space $\mathcal{H}$ or as *linear operators* acting *on* the space $\mathcal{H}$. The latter formulation—standard in physics—shines in the description of quantum systems whose state is not known, as they allow to present a state as a probability distribution of states.

▶ **Example 4.** Intuitively, a state corresponds to a *preparation*, *i.e.* an action performed to set up an experiment. Given two states ρ_0 and ρ_1, there must be a state $r\rho_0 + (1-r)\rho_1$ for any $r \in [0, 1]$. Indeed, a valid way to build a state from two preparations ρ_0 and ρ_1 is the preparation "flip a fair coin: if heads, continue as the preparation ρ_0, otherwise continue as the preparation ρ_1", yielding $\rho = \frac{1}{2}\rho_0 + \frac{1}{2}\rho_1$. A probability distribution of states (such as ρ) also arises naturally to express *partial knowledge* (or *degrees of belief*) about the state of a system. For example, ρ above expresses we do not know if the state of the system is ρ_0 or ρ_1.

Before giving further details, we need to recall some notions from linear algebra.

Notation	Description
z^*	Complex conjugate of the complex number z.
$\|\psi\rangle$	Vector. Also known as a *ket*.
$\langle\psi\|$	Vector dual to $\|\psi\rangle$. Also known as a *bra*.
$\langle\varphi\|\psi\rangle$	Inner product between the vectors $\|\varphi\rangle$ and $\|\psi\rangle$.
$\|\varphi\rangle \otimes \|\psi\rangle$	Tensor product of $\|\varphi\rangle$ and $\|\psi\rangle$.
$\|\varphi\psi\rangle$	Abbreviated notation for $\|\varphi\rangle \otimes \|\psi\rangle$.
ρ^T	Transpose of the matrix ρ.
$\rho^\dagger$	Adjoint (*a.k.a.* Hermitian conjugate, conjugate-transpose) of ρ: $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^\dagger = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix}$.
$\langle\varphi\|\rho\|\psi\rangle$	Inner product between $\|\varphi\rangle$ and $\rho\|\psi\rangle$. Equivalently, inner product between $\rho^\dagger\|\varphi\rangle$ and $\|\psi\rangle$.

Table 1 Main linear algebra notations

Hilbert Spaces. A finite n-dimensional Hilbert space $\mathcal{H}$ is a complex vector space (isomorphic to) $\mathbb{C}^n$ with an inner product $\langle\cdot|\cdot\rangle$. Column vectors are written $|\psi\rangle$, with ψ a label, and their conjugate-transpose $\langle\psi| \stackrel{\text{def}}{=} |\psi\rangle^\dagger$. Table 1 (from [23]) summarizes relevant notations.

Given a Hilbert space $\mathcal{H}$, $\mathcal{L}(\mathcal{H})$ denotes the set of **linear operators** *on* $\mathcal{H}$ (*i.e.* from $\mathcal{H}$ to $\mathcal{H}$), which we identify with their matrix representations. The **adjoint** $\rho^\dagger$ of a matrix ρ is its conjugate-transpose. We denote by $\mathrm{Id}_{\mathcal{L}(\mathcal{H})}$ the identity matrix in $\mathcal{L}(\mathcal{H})$. The **trace** of a square matrix $\rho = (\rho_{ij})_{ij} \in \mathbb{C}^{n\times n}$ is $\mathrm{tr}(\rho) = \sum_i \rho_{ii}$.

A relevant subclass of $\mathcal{L}(\mathcal{H})$ is the set of positive operators $\mathcal{L}(\mathcal{H})^+$, playing the role of "non-negative real numbers". A linear operator ρ is **positive** if $\langle\psi|\rho|\psi\rangle \in \mathbb{R}_0^+$ for any $|\psi\rangle \in \mathcal{H}$.

▶ Remark 5. In the 1-dimensional space $\mathbb{C}^1$, the set of positive matrices in $\mathcal{L}(\mathbb{C}) = \mathbb{C}^{1\times 1} = \mathbb{C}$ is $\mathcal{L}(\mathbb{C})^+ = \mathbb{R}_0^+$.

Quantum states as vectors in $\mathcal{H}$. A common way to represent the states of a quantum system is by unit vectors in $\mathcal{H}$, *i.e.* vectors $|\psi\rangle$ such that $\langle\psi|\psi\rangle = 1$. Considering a qubit, the vectors $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ form an orthonormal basis for its state space $\mathbb{C}^2$: an arbitrary state vector can be written $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Intuitively, the states $|0\rangle$ and $|1\rangle$ of a qubit are analogous to the two values 0 and 1 which a *bit* may take. The way a qubit differs is that *superpositions* (*i.e.* linear combinations) of these two states also exist. Some important states are $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, which form the Hadamard basis.

Quantum states as density operators acting on $\mathcal{H}$. Another usual way in physics to describe the state of a quantum system is by a *positive operator* ρ *with trace* 1—called a **density operator**—that *acts* on the state space $\mathcal{H}$ of the system. We denote by $\mathcal{D}(\mathcal{H}) \subset \mathcal{L}(\mathcal{H})^+$ the set of density operators acting on $\mathcal{H}$. A vector $|\psi\rangle$ is represented by the matrix $|\psi\rangle\langle\psi|$, called a *pure state*. If a quantum system is in the state ρ_i with probability p_i, the density operator for the system is $\sum_i p_i\rho_i$. This formalizes Ex. 4: if we can prepare qubits in pure states, we can also prepare qubits in any probability distribution of states.

Composite systems. The *state space* of a composite physical system is the tensor product of the state spaces of its components. Given several systems, with the states $(\rho_i)_{i\in I}$, the *state* of the composite system is $\bigotimes_{i\in I} \rho_i$. Consider the Hilbert spaces $\mathcal{H}_1$ with $\{|\psi_i\rangle\}_{i\in I}$ one of its bases, and $\mathcal{H}_2$ with $\{|\phi_j\rangle\}_{j\in J}$ one of its bases. Their tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ is the

Hilbert space with basis $\{|\psi_i\rangle \otimes |\phi_j\rangle\}_{(i,j)\in I\times J}$. A short notation for $|\psi\rangle \otimes |\phi\rangle$ is $|\psi\phi\rangle$. A pure quantum state in $\mathcal{H}_1 \otimes \mathcal{H}_2$ is *separable* when it can be expressed as the tensor product of two vectors of $\mathcal{H}_1$ and $\mathcal{H}_2$. Otherwise, it is *entangled*, as for example the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

The **partial trace** $\mathrm{tr}_{\mathcal{L}(\mathcal{H}_1)}$ of an operator in $\mathcal{L}(\mathcal{H}_1 \otimes \mathcal{H}_2)$ is an operator in $\mathcal{L}(\mathcal{H}_2)$ defined as $\mathrm{tr}_{\mathcal{L}(\mathcal{H}_1)}(\rho \otimes \rho') = \mathrm{tr}(\rho)\,\rho'$ for $\rho \in \mathcal{L}(\mathcal{H}_1)$ and $\rho' \in \mathcal{L}(\mathcal{H}_2)$, and extended to the general case by linearity. The partial trace allows to "*forget*" or "*discard*" part of a composite system.

Quantum operations (*a.k.a.* quantum channels). States $\rho \in \mathcal{D}(\mathcal{H})$ transform as $\rho' = \mathcal{E}(\rho)$, where the map $\mathcal{E} : \mathcal{D}(\mathcal{H}) \to \mathcal{D}(\mathcal{H}')$, called **quantum operation** or **quantum channel**, is:

1. *Completely Positive*: $\mathcal{E}$ sends positive operators to positive operators, and do so also when acting on a sub-part of a larger, potentially *entangled*, composite system; *i.e.* any "extension" $\mathrm{Id} \otimes \mathcal{E}$ (with Id the identity) *carries a positive operator to a positive one.*
2. *Trace-Preserving*: if $\mathrm{tr}(\rho) = 1$ then $\mathrm{tr}(\mathcal{E}(\rho)) = 1$.
3. *Convex-Linear*: $\mathcal{E}$ acts on a mixture of states by acting on each component individually, *i.e.* $\mathcal{E}\left(\sum_{i\in I} p_i\rho_i\right) = \sum_{i\in I} p_i\mathcal{E}(\rho_i)$.

By 1 and 2, $\mathcal{E}$ (or more generally $\mathrm{Id} \otimes \mathcal{E}$) sends a density operator to a density operator.

▶ **Example 6.** Two fundamental ways to modify a quantum state are *unitary transformations*, describing the evolution of an isolated system, and *measurements*. A matrix $U \in \mathcal{L}(\mathcal{H})$ is unitary when $UU^\dagger = U^\dagger U = \mathrm{Id}$. The corresponding quantum channel is $\mathcal{E} : \rho \mapsto U\rho U^\dagger$. Similarly, measurement operators M_m act on a state ρ as $M_m\rho M_m^\dagger$. One more example of quantum operation is the partial trace.

Quantum Instruments. Quantum channels describe the evolution of quantum states, *ignoring classical information*—such as measurement outcomes—which plays a central role in "real" (or thought) experiments, as in Fig. 1. A generalization of channels are *quantum instruments*, mathematical devices that model "real" devices by specifying a post-measurement state per classical outcome. This provides a framework where measurements are operations producing both quantum state updates and classical data. A **quantum instrument** is an indexed collection $\{\mathcal{E}_a\}_a$ of completely positive linear maps that sum to a trace preserving map (*i.e.* to a quantum channel). Intuitively, the labels a are the *classical* outcomes ($a^{\mathtt{t}}/a^{\mathtt{f}}$) of a measurement—think $a \in \mathsf{Val}(A)$. An instrument $\{\mathcal{E}_a\}_a$ induces a quantum channel $\mathcal{E}(\rho) = \sum_a \mathcal{E}_a(\rho)$, with $\Pr(a \mid \rho) = \mathrm{tr}(\mathcal{E}_a(\rho))$ providing the outcome probabilities. We refer *e.g.* to [30, page 112] for details.

▶ Remark 7 (Pure quantum and pure classical). The two limit cases yield standard notions.

- Quantum instruments with no classical outcome (*i.e.* with a single label a) are simply quantum operations.
- Quantum instruments from $\mathbb{C}$ to $\mathbb{C}$ are (isomorphic to) probability distrubutions.

2.3 Quantum Bayesian Networks (Instrument-based)

The generalization of Bayesian networks in [15] is based on the notion of quantum instruments. In Sec. 4, we will provide a definition of quantum Bayesian Network in terms of Q-factors—the original definition (to which ours is equivalent) is not necessary to the reader.

In this background section, we briefly review the key ideas in [15], and what are the issues with the semantics there. The causal structure is described by a DAG (as in Fig. 1); to include quantum sources of causality, the set of nodes is extended with new unobserved nodes, corresponding to quantum systems. To each node of the generalized DAG is associated a *family of quantum instruments* rather than a CPT, as we illustrate in Ex. 9.

▶ Remark 8. Recall that to each BN node is associated a CPT, *i.e.* a *family of probability distributions* indexed by values. Intuitively, quantum instruments generalize probability distributions, so a *family of quantum instruments* indexed by values generalizes a CPT.

▶ **Example 9** (Alice's instruments). Referring to Fig. 1, we want to model that Alice (the node A) performs the following operations, depending on the outcome of X. Given x^t, Alice measures the qubit q_1 on the $(|0\rangle, |1\rangle)$ basis, returning a^t if she obtains $|0\rangle$ and a^f if not. Given x^f, Alice measures q_1 on the $(|-\rangle, |+\rangle)$ basis, returning a^t if she obtains $|+\rangle$, a^f if not.

To model this setting, we associate to A a *family* of instruments, indexed by $\mathsf{Val}(X)$: $\{\{\mathcal{E}_{a^t,x^t}, \mathcal{E}_{a^f,x^t}\}, \{\mathcal{E}_{a^f,x^f}, \mathcal{E}_{a^f,x^f}\}\}$. The instrument given x^t is $\{\mathcal{E}_{a^t,x^t}, \mathcal{E}_{a^f,x^t}\}$ (from $\mathcal{L}(\mathcal{H}_{q_1})$ to $\mathbb{C}$) with $\mathcal{E}_{a^t,x^t} : \rho \mapsto |0\rangle\langle 0|\,\rho\,|0\rangle\langle 0|$ and $\mathcal{E}_{a^f,x^t} : \rho \mapsto |1\rangle\langle 1|\,\rho\,|1\rangle\langle 1|$. The instrument given x^f is $\{\mathcal{E}_{a^f,x^f}, \mathcal{E}_{a^f,x^f}\}$, with $\mathcal{E}_{a^t,x^f} : \rho \mapsto |+\rangle\langle +|\,\rho\,|+\rangle\langle +|$ and $\mathcal{E}_{a^f,x^f} : \rho \mapsto |-\rangle\langle -|\,\rho\,|-\rangle\langle -|$.

Semantics of quantum Bayesian networks. The interpretation of a quantum Bayesian network $\mathcal{B}$ over a set of classical variables $\mathbf{X}$ and a set of quantum systems $\mathbf{Q}$ is defined in [15] as a suitable product of the instruments associated to the nodes. Similarly to BNs, the semantics $(\!|\mathcal{B}|\!)$ of $\mathcal{B}$ is a probability distribution over $\mathbf{X}$, reflecting the fact that quantum states cannot be observed directly—we can only observe the classical outcomes of measurements. This is exactly what happens in the Bell set-up (Ex. 1), where we compute $\Pr(a, b \mid x, y)$.

Lack of compositionality. While intuitive and natural, the approach via quantum instruments has a limit in *the lack of compositionality.* Indeed, the interpretation of a quantum Bayesian network $\mathcal{B}$ in [15] needs to follow a specific order (see [15, page 12]). The definition of an instrument does not adapt well to interpreting arbitrary sub-components (as in Fig. 3). Given a partition of $\mathcal{B}$ into $\mathcal{B}_1$ and $\mathcal{B}_2$, even if we can define $(\!|\mathcal{B}_1|\!)$ and $(\!|\mathcal{B}_2|\!)$, we do not have a natural composing function yielding $(\!|\mathcal{B}_1|\!) \bullet (\!|\mathcal{B}_2|\!) = (\!|\mathcal{B}|\!)$. There are two distinct issues.

1. Consider Fig. 3 with its left and right sub-graphs. The left sub-graph should be interpreted by an instrument $\{\mathcal{E}^l_x \mid x \in \mathsf{Val}(X)\}$ with $\mathcal{E}^l_x : \mathcal{H}_2 \to \mathcal{H}_1$, and the right sub-graph by $\mathcal{E}^r : \mathcal{H}_1 \to \mathcal{H}_2$. The interpretation of the full graph is an instrument $\{\mathcal{E}_x \mid x \in \mathsf{Val}(X)\}$ with $\mathcal{E}_x : \mathbb{C} \to \mathbb{C}$. However, the operations on instruments do not immediately yield from $\mathcal{E}^l_x$ and $\mathcal{E}^r$ a function of type $\mathbb{C} \to \mathbb{C}$.
2. A further issue is the impossibility (in general) of dealing by components with hidden variables, exactly as happens for CPTs, which are closed by product but not by sum.

▶ Remark 10 (Marginals and hidden variables). A technical but crucial point when considering compositionality is the variables of interest—or dually, the hidden variables. In Fig. 3, the shaded node Y corresponds to a hidden (or unobserved) variable. What we want here is the marginal $\Pr(x)$, which can be obtained from $\Pr(x, y)$ by summing-out the irrelevant variable: $\sum_{y \in \mathsf{Val}(Y)} \Pr(x, y)$. When computing the semantics of the right component, it would be desirable to already sum-out Y. This kind of operation is regularly performed in Bayesian inference algorithms to reduce the size of computations.

Addressing both issues with compositionality motivates the framework of *quantum factors* that we introduce in the next section. In the classical setting, abstracting CPTs into *factors* gives a status to partial calculations. We will follow a similar path.

3 Quantum Factors

A quantum Bayesian network is a DAG with nodes *a set* $\mathbf{X}$ *of r.v.s* and a *set* $\mathbf{Q}$ *of quantum systems*. We associate to each node a *function*—called quantum factor—from $\mathsf{Val}(\mathbf{X})$ to *positive operators* on the relevant Hilbert space. We then interpret in the same way quantum Bayesian networks, as well as any sub-component. In this section we define quantum factors (Q-factors). In Sec. 4, we reformulate the definition of quantum Bayesian networks in this setting, and explain the equivalence with [15]. Let us illustrate the idea with some examples.

▶ **Example 11.** Consider Ex. 4, the preparation of a single-qubit quantum system Q depending on a classical r.v. X (*e.g.* a fair coin). This is described by the following DAG: $X \rightarrow Q$. The interpretation of this system is a function $\phi : \mathsf{Val}(X) \to \mathcal{L}(\mathcal{H}_Q)^+$ giving for each $x \in \mathsf{Val}(X)$ a positive matrix ρ_x on $\mathcal{H}_Q$. If we are interested only in the state of the qubit, we can sum-out X to obtain the density matrix $\Pr(x^t)\rho_{x^t} + \Pr(x^f)\rho_{x^f}$, as in Ex. 4.

As another example, the preparation of a system Q depending on two classical r.v.s X and Y is a function from $\mathsf{Val}(X, Y)$ to $\mathcal{L}(\mathcal{H}_Q)^+$, yielding four positive matrices $\rho_{xy} \in \mathcal{L}(\mathcal{H}_Q)^+$.

▶ **Example 12** (Alice & Bob). Consider the DAG in Fig. 1, with four classical variables A, B, X and Y, and one quantum node Q producing two entangled qubits q_1 and q_2 in the state space $\mathcal{H}_{q_1} \otimes \mathcal{H}_{q_2}$. We associate to each node a function as follows.

- To X a function $\phi^X : \mathsf{Val}(X) \to \mathcal{L}(\mathcal{H}_\emptyset)^+ = \mathbb{R}_0^+$ (recall Rem. 5). Similarly for Y.
- To Q a positive matrix in $\mathcal{L}(\mathcal{H}_{q_1} \otimes \mathcal{H}_{q_2})^+$.
- To A a function $\phi^A : \mathsf{Val}(X) \times \mathsf{Val}(A) \to \mathcal{L}(\mathcal{H}_{q_1})^+$.

Quantum registers. For simplicity's sake, from now on we assume quantum systems to be composed of qubits. Assuming a countable set $\mathbb{Q}$ of qubits $q_0, q_1, \dots$ with associated 2-dimensional Hilbert spaces $\mathcal{H}_{q_0}, \mathcal{H}_{q_1}, \dots$, a **quantum register** is a *finite* set of qubits $Q \subseteq \mathbb{Q}$. Given a quantum register Q of n qubits, its associated Hilbert space is $\mathcal{H}_Q = \bigotimes_{q\in Q} \mathcal{H}_q \cong \mathbb{C}^{2^n}$. We write $Q \uplus Q'$ the *disjoint union* of the registers Q and Q', only defined when $Q \cap Q' = \emptyset$.

Quantum Factors. Please compare (3) below with (2) the definition of a factor in Sec. 2.1.

▶ **Definition 13** (Quantum Factor (Q-factor)). *Let* $\mathbf{X}$ *be a set of random variables and* Q *a quantum register. A* ***quantum factor*** *(shortened as* ***Q-factor****)* ϕ *over* $(\mathbf{X}, Q)$ *is a function*

$$\phi : \mathsf{Val}(\mathbf{X}) \to \mathcal{L}(\mathcal{H}_Q)^+ \tag{3}$$

mapping each tuple of values $\mathbf{x}$ *to a* positive *operator in* $\mathcal{L}(\mathcal{H}_Q)$. *The* ***scope*** *of* ϕ *is* $\mathbf{X} \cup Q$.

▶ Remark 14 (Pure classical and pure quantum). The two limit cases yield familiar notions.

- If Q is empty, then positive operators in $\mathcal{L}(\mathbb{C}^1)$ are exactly elements of $\mathbb{R}_0^+$, yielding the standard definition of a factor over a set of random variables.
- If $\mathbf{X}$ is empty, then a Q-factor simply corresponds to *a positive operator* in $\mathcal{L}(\mathcal{H}_Q)^+$.

The trivial Q-factor over $(\emptyset, \emptyset)$ simply corresponds to a *non-negative* real. We denote by $\mathbf{1}$ the trivial Q-factor corresponding to 1.

3.1 Product and Sum of Quantum Factors

We show that Q-factors admit and are *closed* under *product* $\odot$ and *summing-out* $\sum$ operations. Remarkably, Q-factors encompasses both tensor networks and factors from Bayesian networks:

- For Q-factors ϕ and ϕ' over classical variables only, $\phi \odot \phi'$ is the product of factors and $\sum_X \phi$ is summing-out (*a.k.a. marginalization*).
- For Q-factors ϕ and ϕ' over quantum registers only, $\phi \odot \phi'$ is the product of tensor networks and $\sum_Q \phi$ is tracing-out (*i.e.* taking the *partial trace* over $\mathcal{H}_Q$).

Sum-out for Q-factors. *Summing out a variable* from a Q-factor corresponds to *marginalization* for classical variables, and to the *partial trace* for quantum systems.

▶ **Definition 15** (Sum-out). *Let ϕ be a Q-factor over $(\mathbf{X}, Q)$.*

- *For $Z \in \mathbf{X}$, the **sum** $\sum_Z \phi$ is the Q-factor over $(\mathbf{X} - \{Z\}, Q)$ defined by:*

$$\left(\sum_Z \phi\right)(\mathbf{y}) \overset{def}{=} \sum_{z \in \mathsf{Val}(Z)} \phi(\mathbf{y}, z) \qquad \textit{for } \mathbf{y} \in \mathsf{Val}(\mathbf{X} - \{Z\})$$

- *For $Q' \subseteq Q$, the **sum** $\sum_{Q'} \phi$ is the Q-factor over $(\mathbf{X}, Q - Q')$ defined by:*

$$\left(\sum_{Q'} \phi\right)(\mathbf{x}) \overset{def}{=} \mathrm{tr}_{\mathcal{L}(\mathcal{H}_{Q'})}(\phi(\mathbf{x})) \qquad \textit{for } \mathbf{x} \in \mathsf{Val}(\mathbf{X})$$

The definition of the product is more delicate, because classical variables and quantum systems behave differently: classical values can be “shared”, while quantum states are linear resources. To mathematically conciliate the two, we describe the product in terms of bases.

Describing Q-factors in terms of the canonical basis. For a qubit q_i, we write $\mathsf{Basis}_{q_i} = (e_i^{00}, e_i^{01}, e_i^{10}, e_i^{11})$ for the canonical orthonormal basis of $\mathcal{L}(\mathcal{H}_{q_i}) \cong \mathbb{C}^{2\times 2}$, *i.e.* $e_i^{00} = |0\rangle\langle 0|$, $e_i^{01} = |0\rangle\langle 1|$, $e_i^{10} = |1\rangle\langle 0|$, and $e_i^{11} = |1\rangle\langle 1|$. Given a quantum register $Q = \{q_1, \ldots, q_m\}$, we write $\mathsf{Basis}_Q = \prod_{i=1}^m \mathsf{Basis}_{q_i}$, which is an orthonormal basis for $\mathcal{L}(\mathcal{H}_Q) = \mathcal{L}(\bigotimes_{i=1}^m \mathcal{H}_{q_i})$. A Q-factor ϕ over $(\mathbf{X}, Q)$ is then associated to the following function, that we denote by $\widehat{\phi}$:

$$\begin{aligned} \widehat{\phi} : \mathsf{Val}(\mathbf{X}) \times \mathsf{Basis}_Q &\to \mathbb{C} \\ (x_1, \ldots, x_n, e_1, \ldots, e_m) &\mapsto \phi(x_1, \ldots, x_n)_{e_1, \ldots, e_m} \end{aligned}$$

where $\phi(x_1, \ldots, x_n)_{e_1, \ldots, e_m}$ is the scalar at position $(e_1, \ldots, e_m)$ of the matrix $\phi(x_1, \ldots, x_n)$. Notice that $\widehat{\phi}$ uniquely determines ϕ, and vice-versa.

▶ **Example 16.** Consider r.v.s X_1 and X_2, and a quantum register $\{q_1, q_2\}$. An example of Q-factor ϕ over $(\{X_1, X_2\}, \{q_1, q_2\})$ is:

$$\phi(x_1^t, x_2^t) = \phi(x_1^t, x_2^f) = \phi(x_1^f, x_2^t) = \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix} \qquad \phi(x_1^f, x_2^f) = \begin{bmatrix} 2&0&0&1 \\ 0&2&0&0 \\ 0&0&2&-1 \\ 1&0&-1&2 \end{bmatrix}$$

We also see ϕ as a function $\widehat{\phi} : \mathsf{Val}(X_1) \times \mathsf{Val}(X_2) \times \mathsf{Basis}_{q_1} \times \mathsf{Basis}_{q_2} \to \mathbb{C}$ sending (x_1, x_2, e_1, e_2) to the scalar corresponding to $e_1 \otimes e_2$. For instance:

- $\widehat{\phi}(x_1^t, x_2^t, e_1^{00}, e_2^{00}) = \ \ 1$, the entry of $\phi(x_1^t, x_2^t)$ corresponding to $e_1^{00} \otimes e_2^{00}$;
- $\widehat{\phi}(x_1^f, x_2^f, e_1^{11}, e_2^{01}) = -1$, the entry of $\phi(x_1^f, x_2^f)$ corresponding to $e_1^{11} \otimes e_2^{01}$.

Product of Q-factors. We now define the product. It shares the values of classical variables (exactly as happens in the factors product, as discussed in Sec. 2.1), while on quantum registers it behaves as the contraction of tensor networks. Below, we use $\widehat{\phi}$ instead of ϕ; the symmetric difference $S \,\Delta\, R \overset{\text{def}}{=} (S \cup R) - (S \cap R)$ replaces the union in the quantum case.

▶ **Definition 17** (Product of Q-factors). *Let ϕ_1 and ϕ_2 be Q-factors respectively over $(\mathbf{X}_1, Q_1)$ and $(\mathbf{X}_2, Q_2)$. The **product** $\phi_1 \odot \phi_2$ is a Q-factor over $(\mathbf{X}_1 \cup \mathbf{X}_2, Q_1 \,\Delta\, Q_2)$ defined as follows. Let $\mathbf{x} \in \mathsf{Val}(\mathbf{X}_1 \cup \mathbf{X}_2)$ and $(\mathbf{e}_1, \mathbf{e}_2) \in \mathsf{Basis}_{Q_1 \Delta Q_2} = \mathsf{Basis}_{Q_1 - Q_2} \times \mathsf{Basis}_{Q_2 - Q_1}$. Call $\mathbf{x}_1$ the elements of $\mathbf{x}$ belonging to $\mathsf{Val}(\mathbf{X}_1)$, and similarly $\mathbf{x}_2$ those in $\mathsf{Val}(\mathbf{X}_2)$. We set*

$$(\widehat{\phi_1 \odot \phi_2})(\mathbf{x}, \mathbf{e}_1, \mathbf{e}_2) \overset{def}{=} \sum_{\mathbf{e}_3 \in \mathsf{Basis}_{Q_1 \cap Q_2}} \widehat{\phi_1}(\mathbf{x}_1, \mathbf{e}_1, \mathbf{e}_3)\ \widehat{\phi_2}(\mathbf{x}_2, \mathbf{e}_2, \mathbf{e}_3)$$

Properties of sum and product. Q-factors are *closed under sum and product.* Easy calculations show that the product is (essentially) associative, product and sum are both commutative, and—crucially—they distribute under suitable conditions, like classical factors.

▶ **Proposition 18.** *Q-factors are* closed *under both product and sum. Moreover:*
1. $(\phi_1 \odot \phi_2) \odot \phi_3 = \phi_1 \odot (\phi_2 \odot \phi_3)$ *provided no qubit appears in the scope of all three Q-factors*
2. $\phi_1 \odot \phi_2 = \phi_2 \odot \phi_1$
3. $\sum_U \sum_V \phi_1 = \sum_V \sum_U \phi_1$
4. $\sum_V (\phi_1 \odot \phi_2) = (\sum_V \phi_1) \odot \phi_2$ *when V is not in the scope of ϕ_2*

(Proof in Appendix B.) The side condition in point 1 is always satisfied by quantum BNs.

4 Quantum Bayesian Networks (Q-factor-based)

We define quantum Bayesian networks in terms of Q-factors. Our definition is equivalent to the instrument-based definition in [15], however moving to Q-factors allows the semantics to address compositionality of sub-networks. To each node, we associate a special kind of Q-factor, called a Q-cpt. This ensures that the semantics of the full network is exactly a probability distribution, and not any arbitrary function with values in $\mathbb{R}_0^+$.

Quantum CPTs. We define a class of Q-factors which plays a similar role to that of CPTs.

▶ **Definition 19** (Quantum CPT (Q-cpt))**.** *A Q-factor ϕ over $(\mathbf{X}, Q)$ is*
- *a **Q-cpt for** $Z \in \mathbf{X}$ **given** $(\mathbf{Y}, Q)$ if $\mathbf{X} = \{Z\} \uplus \mathbf{Y}$, and $\forall \mathbf{y} \in \mathsf{Val}(\mathbf{Y})$:*

$$\left(\sum_Z \phi\right)(\mathbf{y}) = \mathrm{Id}_{\mathcal{L}(\mathcal{H}_Q)}$$

- *a **Q-cpt for** $Q' \subseteq Q$ **given** $(\mathbf{X}, Q'')$ if $Q = Q' \uplus Q''$, and $\forall \mathbf{x} \in \mathsf{Val}(\mathbf{X})$:*

$$\left(\sum_{Q'} \phi\right)(\mathbf{x}) = \mathrm{Id}_{\mathcal{L}(\mathcal{H}_{Q''})}$$

The two limit cases correspond to standard, familiar notions.
- In the *pure classical case* ($Q = \emptyset$), being a Q-cpt amounts to being a CPT. Indeed, a Q-cpt for X given $(\mathbf{Y}, \emptyset)$ is a Q-factor ϕ over $(\{X\} \uplus \mathbf{Y}, \emptyset)$ which satisfies $\sum_{x \in \mathsf{Val}(X)} \phi(x, \mathbf{y}) = 1$.
- In the *pure quantum case* ($\mathbf{X} = \emptyset$), being a Q-cpt amounts to being *Choi state*, a well-known condition on operators that we will detail in Sec. 4.2.

Quantum Bayesian Networks. We define quantum Bayesian networks as DAGs with a Q-cpt for each node.

▶ **Definition 20** (Quantum Bayesian Network (QBN))**.** *A **quantum Bayesian network** over a set $\mathbf{X}$ of r.v.s and a set $\mathbf{Q}$ of disjoint non-empty quantum registers is a pair $(\mathcal{G}, \Phi)$ where:*
- *$\mathcal{G}$ is a directed acyclic graph over the set of nodes $\mathbf{X} \cup \mathbf{Q}$;*
- *each out-edge of a node $Q \in \mathbf{Q}$ is labelled by a non-empty register Q_i, with $\uplus_i Q_i = Q$;*
- *each out-edge of a node $X \in \mathbf{X}$ is labelled by X;*
- *Φ assigns to each $V \in \mathbf{X} \cup \mathbf{Q}$ a Q-cpt for V given the r.v.s and qubits labelling its in-edges.*

▶ **Example 21** (Alice's Q-cpt)**.** Consider Fig. 1 with the setting of Ex. 9. The node A is given the Q-factor $\phi^A : \mathsf{Val}(A) \times \mathsf{Val}(X) \to \mathcal{L}(\mathcal{H}_{q_1})$ below, that is a Q-cpt for A given $(\{X\}, \{q_1\})$:

$$\begin{array}{lcl@{\qquad}lcl}
(a^t, x^t) & \mapsto & |0\rangle\langle 0| = \left[\begin{smallmatrix}1 & 0\\ 0 & 0\end{smallmatrix}\right] & (a^f, x^t) & \mapsto & |1\rangle\langle 1| = \left[\begin{smallmatrix}0 & 0\\ 0 & 1\end{smallmatrix}\right] \\
(a^t, x^f) & \mapsto & |+\rangle\langle +| = \frac{1}{2}\left[\begin{smallmatrix}1 & 1\\ 1 & 1\end{smallmatrix}\right] & (a^f, x^f) & \mapsto & |-\rangle\langle -| = \frac{1}{2}\left[\begin{smallmatrix}1 & -1\\ -1 & 1\end{smallmatrix}\right]
\end{array}$$

4.1 Semantics and Compositionality

The semantics of a QBN is a probability distribution, defined akin to that of a BN (Thm. 2).

▶ **Definition 22.** *For $\mathcal{B} = (\mathcal{G}, \Phi)$ a QBN over r.v.s $\mathbf{X}$ and registers $\mathbf{Q}$, its **semantics** is:*

$$(\!|\mathcal{B}|\!) \stackrel{def}{=} \bigodot_{V \in \mathbf{X} \cup \mathbf{Q}} \Phi(V)$$

▶ **Proposition 23.** *With the above notations, $(\!|\mathcal{B}|\!)$ is a probability distribution over $\mathbf{X}$.*

The proof is obtained in a similar way to that for the classical result [24].

The formalism allows us to naturally interpret sub-components of a QBN, as in Fig. 3. Call $(\mathcal{G}', \Phi')$ a **sub-network** of a QBN $(\mathcal{G}, \Phi)$ when $\mathcal{G}'$ is a sub-graph of $\mathcal{G}$ and Φ' is the restriction of Φ to nodes of $\mathcal{G}'$. The semantics of a sub-network is defined as for the full network: the product of its Q-factors. Compositionality follows from associativity of $\odot$.

▶ **Proposition 24** (Compositionality)**.** *Let $\mathcal{B} = (\mathcal{G}, \Phi)$ be a QBN, and $\mathcal{B}_1 = (\mathcal{G}_1, \Phi_1)$ and $\mathcal{B}_2 = (\mathcal{G}_2, \Phi_2)$ a partition in two sub-networks (*i.e. *$\mathcal{G} = \mathcal{G}_1 \uplus \mathcal{G}_2$). Then:*

$$(\!|\mathcal{B}|\!) = (\!|\mathcal{B}_1|\!) \odot (\!|\mathcal{B}_2|\!)$$

4.2 Equivalence with Instrument-based Quantum Bayesian Networks

Choi states and Q-cpts. The **Choi-Jamiolkowski isomorphism** (see *e.g.* [30]) is a classical, well-known result allowing to see linear maps from $\mathcal{L}(\mathcal{H}_1)$ to $\mathcal{L}(\mathcal{H}_2)$ as matrices in $\mathcal{L}(\mathcal{H}_1 \otimes \mathcal{H}_2)$. It has many applications, basic ones being to check easily whether two quantum operations are equal—by checking their images are the same—or whether a map is completely positive; we refer to [30] for details. Let us denote by $\mathcal{J}$ the Choi-Jamiolkowski isomorphism. Given a quantum operation $\mathcal{E} : \mathcal{L}(\mathcal{H}_1) \to \mathcal{L}(\mathcal{H}_2)$, complete positivity of $\mathcal{E}$ is equivalent to positivity of $\mathcal{J}(\mathcal{E})$, and trace-preserving of $\mathcal{E}$ is equivalent to $\mathrm{tr}_{\mathcal{L}(\mathcal{H}_2)}(\mathcal{J}(\mathcal{E})) = \mathrm{Id}_{\mathcal{L}(\mathcal{H}_1)}$. Hence, operators $\rho \in \mathcal{L}(\mathcal{H}_1 \otimes \mathcal{H}_2)$ that are positive and such that $\mathrm{tr}_{\mathcal{L}(\mathcal{H}_2)}(\rho) = \mathrm{Id}_{\mathcal{L}(\mathcal{H}_1)}$ are exactly the images by $\mathcal{J}$ of quantum operations, and are called **Choi states**. Please notice that being a Q-cpt for a quantum register Q', *i.e.* the second item of Def. 19, corresponds to being a Choi state (once values for the r.v.s are fixed) with $\left(\sum_{Q'} \phi\right)(\mathbf{x}) = \mathrm{tr}_{\mathcal{L}(\mathcal{H}_{Q'})}(\phi(\mathbf{x})) = \mathrm{Id}_{\mathcal{L}(\mathcal{H}_{Q''})}$—the positivity requirement being already part of the definition of a Q-factor.

Equivalence with Instrument-based Quantum Bayesian Networks. Our definition of quantum Bayesian networks is equivalent to the instrument-based one of [15]. Indeed, the definition of the DAGs is the same, and it is straightforward to provide a bijection between a Q-cpt associated to a node in our formulation, and instruments associated to the same node in [15]. The proof is similar to that for the Choi-Jamiolkowski isomorphism (see Appendix C). The composition of instruments in [15] corresponds to the product of Q-cpts.

5 Quantum (Bayesian) Proof-Nets

In this section, we introduce quantum proof-nets. They can be thought of as a typed version of quantum Bayesian networks, built on the graph formalism of Multiplicative Linear Logic. Any quantum Bayesian network can be encoded as a quantum proof-net and, conversely, to any quantum proof-net of appropriate type is associated a quantum Bayesian network. What we gain by moving to the typed setting of proof-nets is:

- This formalism encodes QBNs and enables formalizing (and working with) sub-networks.
- Typing allows us to *modularly compose* quantum proof-nets of compatible interfaces.
- Finally, we have *a linear form of high-order*, as already sketched in Fig. 4 where the network $\mathcal{N}_1$ of type $(A \multimap C) \multimap D$ is expecting as input a network of type $(A \multimap C)$.

Formulas. We assume given two countable sets of **names** denoted by metavariables $X, Y, \ldots$ and $q_1, q_2, \ldots$ which we call respectively **classical names** and **q-names**. Formulas are those of the Multiplicative fragment of Linear Logic (MLL):

$$F, G ::= X^+ \mid X^- \mid q^+ \mid q^- \mid F \otimes G \mid F \parr G$$

Negation $(\cdot)^\perp$ is defined inductively by $(X^+)^\perp \stackrel{\text{def}}{=} X^-$, $(X^-)^\perp \stackrel{\text{def}}{=} X^+$, $(q^+)^\perp \stackrel{\text{def}}{=} q^-$, $(q^-)^\perp \stackrel{\text{def}}{=} q^+$, $(F \otimes G)^\perp \stackrel{\text{def}}{=} F^\perp \parr G^\perp$ and $(F \parr G)^\perp \stackrel{\text{def}}{=} F^\perp \otimes G^\perp$. **Linear arrow** is defined as usual by $F \multimap G \stackrel{\text{def}}{=} F^\perp \parr G$. We reserve the notation Q^+ for formulas of the shape $\otimes_{i \in I} q_i^+$. Capital Greek letters $\Gamma, \Delta, \ldots$ vary over finite sequences of formulas, *i.e.* $\Delta = F_1, \ldots, F_n$. By $\mathsf{Nm}(F)$ we denote the set of all names which appear in the formula F, and similarly for $\mathsf{Nm}(\Delta)$. For example, $\mathsf{Nm}(X^- \parr (Y^+ \otimes X^+), q^+ \parr q^-) = \{X, Y, q\}$.

A key property of formulas in Linear Logic is (positive/negative) **polarity**, which we indicate for atoms and extend to formulas as follows, defining (strictly) polarized formulas:

Positive formulas: $P, P' ::= X^+ \mid q^+ \mid P \otimes P'$
Negative formulas: $N, N' ::= X^- \mid q^- \mid N \parr N'$

▶ Remark 25. Dual polarities carry a connotation of output/input, answer/question, active/passive, player/opponent, etc. Negative atoms are seen as *inputs*, and positive ones as *outputs*. An intuition—at the base of the Linear Logic interpretation of the information flow—is that information travels upwards (resp. downwards) on negative (resp. positive) atoms.

MLL proof-nets (with boxes). In Linear Logic, proofs admit both a *sequent calculus* syntax in the form of **proof-trees** and a *graph syntax*, in the form of **proof-net**. Here, we are interested in the latter. We omit the sequent calculus presentation—which is straightforward to deduce for the reader familiar with Linear Logic, but not relevant for our purpose.

▶ **Definition 26** (Proof-Nets). *We call **typed** a labelled (partial*[2]*) graph $\mathcal{R}$ built from the alphabet of nodes on Fig. 5, with edges labelled by* MLL *formulas. Edges incident to a node n need to respect the typing conditions of the grammar, and are classified either as **premises** of n (depicted above n) or as its **conclusions** (depicted below n). We further require that an edge is the conclusion of at most one node and the premise of at most one node.*

- ***Proof-nets.*** *A typed graph $\mathcal{R}$ is a **proof-net** if it has no pending premises (*i.e.* every edge in $\mathcal{R}$ is the conclusion of some node) and if it is **correct**: every cycle in $\mathcal{R}$ uses the two premises of a same $\parr$-node or the two premises of a same* c*-node.*
- ***Conclusions.*** *The labels of the edges which are premise of no node (*a.k.a.* pending conclusions) are called the **conclusions** of $\mathcal{R}$. The **type** of a proof-net $\mathcal{R}$ of conclusions $\Delta = C_1, \ldots, C_k$ is the sequent $\vdash \Delta$. We also write $\mathcal{R} \vdash \Delta$.*

▶ **Example 27.** The typed graph in Fig. 2—where $Q^+ = q_1^+ \otimes q_2^+$—is a proof-net.

[2] A partial graph may have *pending edges*, which may be either out-edges (as in Fig. 2) or in-edges.

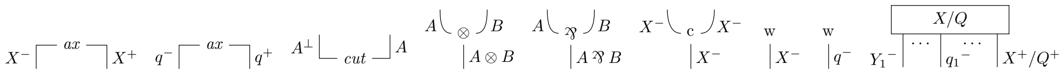


Figure 5 Grammar of nodes for proof-nets; atoms in the conclusions of a box are pairwise distinct

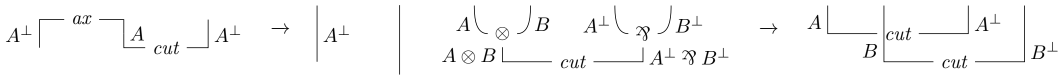


Figure 6 Reduction rules of proof-nets

The grammar in Fig. 5 extends the standard nodes of MLL with a new node, called a **box**, which has *exactly one positive conclusion*—its **output**—of type either X^+ or $Q^+ = \otimes_{i \in I} q_i^+$. The negative conclusions—called **inputs**—are atomic. All atoms appearing in the conclusions of a box are pairwise distinct. The grammar is similar to that introduced in [11]. However, while in [11] to each box is associated a CPT, here we associate a Q-cpt.

Quantum proof-nets. As before, we identify each name X with the r.v. $(X, \{x^{\mathtt{t}}, x^{\mathtt{f}}\})$ and each q-name q_i with a distinct qubit. With this implicit assumption, if P is the output of a box $\mathtt{b}$, then $\mathsf{Nm}(P)$ is either a random variable $\{X\}$ or a quantum register $\{q_1, \ldots, q_k\}$.

▶ **Definition 28** (Quantum proof-nets). *A* ***quantum proof-net*** *(qpn) is a pair* $(\mathcal{R}, \Phi)$ *where:*
1. $\mathcal{R}$ *is a proof-net such that all the atoms appearing in the outputs of distinct boxes are pairwise distinct, and all the negative q-atoms which are inputs of distinct boxes are pairwise distinct.*
2. Φ *assigns to each box* $\mathtt{b}$ *of output* P *a Q-cpt* $\Phi(\mathtt{b})$*, for* $\mathsf{Nm}(P)$ *given the (classical and q-) names of the inputs of* $\mathtt{b}$*.*

Thanks to point 1, we denote a box in a qpn $\mathcal{R}$ by its positive conclusion—*e.g.* $\mathtt{b}^P$ for the unique box of output P. When Φ is irrelevant, we often write "a qpn $\mathcal{R}$" for "a qpn $(\mathcal{R}, \Phi)$".

Reduction rules & Normal forms. Quantum proof-nets are equipped with the standard reduction rules for MLL proof-nets, presented in Fig. 6. These rules define a binary relation $\to$ on qpns, written $\mathcal{R} \to \mathcal{R}'$ (read $\mathcal{R}$ reduces to $\mathcal{R}'$). A qpn $\mathcal{R}$ is in **normal** form (or just normal) if there is no $\mathcal{R}'$ such that $\mathcal{R} \to \mathcal{R}'$. Please notice that, because of boxes, a proof-net in normal form can still contain *cut*-nodes—see *e.g.* the proof-net in Fig. 2.

▶ **Proposition 29.** *The reduction* $\to$ *preserves both the correctness and the conclusions of a proof-net, is terminating and is confluent.*

5.1 Quantum Bayesian Networks and Quantum Proof-Nets

We now relate the two formalisms, quantum Bayesian networks and quantum proof-nets.

Polarized proof-nets as directed graphs. Polarities embed a notion of orientation (Rem. 25), allowing to read a typed graph as a DAG. Restricting our attention to proof-nets whose edges are labelled only by polarized formulas, known as **polarized proof-nets** [21], to each edge is associated a *direction*: upwards for edges labelled by a negative formula, downwards for edges labelled by a positive formula. The following is well-known for polarized proof-nets.

▶ **Lemma 30** (Polarized correctness [20]). *Let* $\mathcal{R}$ *be a typed graph (Def. 26) whose edges are labelled only by polarized formulas. The graph* $\mathcal{R}$ *is* correct *if and only if orienting the edges of* $\mathcal{R}$ *according to their polarity yields a DAG.*

By polarized correctness, every polarized proof-net $\mathcal{R}$ can be seen as a DAG where edges are oriented according to their polarity. This induces a DAG on the boxes $\mathsf{Boxes}(\mathcal{R}) = \{\mathsf{b}^{P_1}, \ldots, \mathsf{b}^{P_n}\}$ of $\mathcal{R}$, hence a DAG $\mathcal{G}_{\mathcal{R}}$ on $\mathsf{Nm}(P_1), \ldots, \mathsf{Nm}(P_n)$: we have a directed edge $\mathsf{Nm}(P_i) \to \mathsf{Nm}(P_j)$ in $\mathcal{G}_{\mathcal{R}}$ whenever there is a directed path from b^{P_i} to b^{P_j} in $\mathcal{R}$.

From quantum proof-nets to QBN's. We call **closed** a quantum proof-net $\mathcal{R} \vdash \Delta$ where all the atoms appearing (as sub-formulas) in Δ are *positive and classical.*

▶ **Lemma 31.** *To each* polarized *qpn* $(\mathcal{R}, \Phi_{\mathcal{R}})$ *of boxes* $\mathsf{Boxes}(\mathcal{R}) = \{\mathsf{b}^{P_1}, \ldots, \mathsf{b}^{P_n}\}$ *is associated a pair* $(\mathcal{G}, \Phi_{\mathcal{G}})$ *with* $\mathcal{G}$ *the DAG induced by* $\mathcal{R}$ *on* $\mathsf{Nm}(P_1), \ldots, \mathsf{Nm}(P_n)$ *and* $\Phi_{\mathcal{G}}(\mathsf{Nm}(P)) \stackrel{def}{=} \Phi_{\mathcal{R}}(\mathsf{b}^P)$*. If* $\mathcal{R}$ *is closed, the pair* $(\mathcal{G}, \Phi_{\mathcal{G}})$ *is a quantum Bayesian network.*

▶ **Example 32.** The proof-net in Fig. 2 is polarized and closed; its DAG is the one in Fig. 1.

From QBNs to quantum proof-nets. A quantum Bayesian network can be encoded as a quantum proof-net rather directly. To a QBN $(\mathcal{G}, \Phi)$, it is straightforward to associate *a typed graph* $\mathcal{N}_{\mathcal{G}}$ as sketched in Figs. 1 and 2. It is immediate that $\mathcal{N}_{\mathcal{G}}$ is polarized and closed. The only delicate point is checking $\mathcal{N}_{\mathcal{G}}$ is correct, hence a proof-net: it follows from Lemma 30, as $\mathcal{G}$ is a DAG. The pair $(\mathcal{N}_{\mathcal{G}}, \Phi)$ clearly satisfies all conditions in Def. 28, making it a qpn.

Every pn has a polarized core. The encoding of Quantum Bayesian networks into polarized closed quantum proof-nets is sound and complete. In fact, we can associate a quantum Bayesian network to any closed quantum proof-net (first reducing it to its normal form).

▶ **Proposition 33** (Polarized core of a Normal Form)**.** *Every* normal *proof-net* $\mathcal{R} \vdash F$ *consists of a polarized proof-net* $Pol(\mathcal{R})$ *on top of the formula tree of* F.

This generalizes in the obvious way to $\mathcal{R} \vdash \Delta$. For example, consider the proof-nets of Fig. 7: each has a formula tree drawn with dashed edges below its polarized core with solid edges.

Putting everything together, we can associate a QBN to (the normal form of) any closed qpn. In the next section, we prove they have the *same semantics* (Thm. 38).

5.2 Compositionality and Modularity

We have now built all the ingredients to satisfy the desiderata outlined in the introduction.

Denotational Semantics. We denote by $\mathsf{Nm}(\mathcal{R})$ the set of all names appearing in $\mathcal{R}$.

▶ **Definition 34** (Semantics of a quantum proof-net)**.** *Let* $(\mathcal{R}, \Phi)$ *be a qpn. Its semantics is:*

$$[\![\mathcal{R}]\!] \stackrel{def}{=} \sum_{\mathsf{Nm}(\mathcal{R})-\mathsf{Nm}(\Delta)} \left(\bigodot_{\mathsf{b} \in \mathsf{Boxes}(\mathcal{R})} \Phi(\mathsf{b}) \right) \quad \textit{where } \Delta \textit{ is the conclusions of } \mathcal{R}.$$

▶ Remark 35. If $\mathcal{R}$ is such that $\mathsf{Boxes}(\mathcal{R}) = \emptyset$, then $[\![\mathcal{R}]\!]$ is the trivial factor $\mathbf{1}$.

The invariance of the semantics via reduction is immediate, because both the boxes and the conclusions are invariants of the reduction.

▶ **Proposition 36** (Invariance)**.** *Let* $\mathcal{R}$ *be a quantum proof-net.*

1. *If* $\mathcal{R} \to \mathcal{R}'$ *then* $\mathcal{R}'$ *is a quantum proof-net with* $[\![\mathcal{R}]\!] = [\![\mathcal{R}']\!]$.
2. *Let* $\mathcal{R}'$ *be the normal form of* $\mathcal{R}$*, then* $[\![\mathcal{R}]\!] = [\![\mathcal{R}']\!] = [\![Pol(\mathcal{R}')]\!]$.

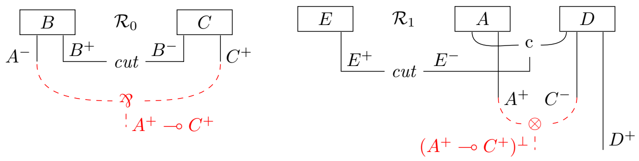


Figure 7 Modularity through typing

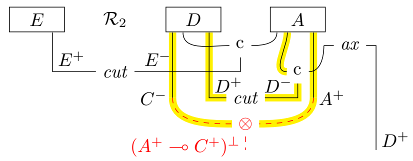


Figure 8 Non-example of proof-net

The crucial question is *compositionality*. We prove that the semantics of a qpn $\mathcal{R}$ can be defined compositionally for any decomposition of $\mathcal{R}$ in sub-graphs.

▶ **Theorem 37** (Compositionality)**.** *Let $(\mathcal{R}, \Phi)$ be a quantum proof-net, decomposed into two sub-graphs $\mathcal{R}_1$ and $\mathcal{R}_2$ with the obvious Q-cpts assignments given by restriction of Φ. Then:*

$$[\![\mathcal{R}]\!] = \sum_{(\mathsf{Nm}(\Delta_1) \cup \mathsf{Nm}(\Delta_2)) - \mathsf{Nm}(\Delta)} ([\![\mathcal{R}_1]\!] \odot [\![\mathcal{R}_2]\!])$$

where Δ (resp. Δ_1, Δ_2) is the conclusions of $\mathcal{R}$ (resp. $\mathcal{R}_1$, $\mathcal{R}_2$).

The proof is similar to [8], adapting to a graph setting the argument from [12].

Soundness and Completeness. In Sec. 5.1, we proved that the formalism of quantum proof-nets is sound and complete *w.r.t.* QBNs. The semantics of a qpn and of the associated QBN are the same, essentially by definition.

▶ **Theorem 38** (Completeness and Soundness with respect to QBNs)**.**

- *To every QBN $\mathcal{B}$ is associated a closed qpn $\mathcal{R}_\mathcal{B}$ with $(\!|\mathcal{B}|\!) = [\![\mathcal{R}_\mathcal{B}]\!]$.*
- *To every* closed *qpn $\mathcal{R} \vdash \Delta$ is associated a QBN $\mathcal{B}_\mathcal{R}$ with $[\![\mathcal{R}]\!] = \sum_{\mathsf{Nm}(\mathcal{R}) - \mathsf{Nm}(\Delta)} (\!|\mathcal{B}_\mathcal{R}|\!)$.*

On Types and Modularity. The typing discipline enables modularity: two qpns of dual type F and $F^\perp$ are *guaranteed* to compose well. More generally, composition is defined by the standard *cut*-rule of sequent calculus—we say that $\mathcal{R}_1, \mathcal{R}_2$ below have **compatibles types**:

$$\frac{\mathcal{R}_1 \vdash \Gamma, F \quad \mathcal{R}_2 \vdash \Delta, F^\perp}{\mathcal{R} \vdash \Gamma, \Delta}\ cut$$

where $\mathcal{R}$ is obtained by connecting the conclusions (labelled by) F and $F^\perp$ with a *cut*-node.

▶ **Proposition 39.** *If the qpns $\mathcal{R}_1$ and $\mathcal{R}_2$ have compatible types, their composition is a qpn. Plugging together the DAGs respectively associated to $\mathcal{R}_1$ and $\mathcal{R}_2$ produces a DAG.*

Fig. 7 revisits the motivational example of Fig. 4 in our typed setting. Fig. 8 presents a non-example of typing for $\mathcal{N}_3$ in Fig. 4: this graph is not a proof-net due to the highlighted cycle. Types enable composition, by acting as an interface which encodes (via $\otimes$ and $⅋$) the DAG's structure.

6 Conclusion and Related Work

This paper brings compositional principles and a typing discipline in the setting of Quantum Bayesian networks, addressing the lack of compositionality and modularity in the original setting with methods from denotational semantics and proof theory. Our framework is fully compatible with classical Bayesian networks and Bayesian inference. The crucial notion is that of *quantum factors*, that directly generalize factors from the theory of (classical) Bayesian networks and satisfy the same key properties for product and sum.

Classical factor over r.v.s $\mathbf{X}$	Quantum factor over r.v.s $\mathbf{X}$ and the quantum register Q
$\phi : \mathsf{Val}(\mathbf{X}) \to \mathbb{R}_0^+$	$\phi : \mathsf{Val}(\mathbf{X}) \to \mathcal{L}(\mathcal{H}_Q)^+$

Their development is highly non-trivial, and is our main technical contribution. A major challenge has been to conciliate the behaviour of classical factors with quantum no-cloning. Our product of Q-factors shares the value of classical variables (as factors do) and coincides with tensor contraction for quantum registers, being fully compatible with both settings.

The framework enables computing the probability distribution associated to a network by multiplying its Q-factors in any arbitrary order (not necessarily in a top-down fashion as in [15]), and by summing out irrelevant variables on sub-components.

Furthermore, we propose a typed formalism for quantum Bayesian networks—based on proof-nets—that guarantees *modularity* through a proof-theoretic approach. The *exact* correspondence between closed proof-nets and QBNs is guaranteed by soundness and completeness results. Finally, it is worth noting that Q-factors are a model of Multiplicative Linear Logic.

Related work. Our semantics is strongly inspired by Selinger's [26]—the graph of the function defining a *quantum factor* can be seen as an *indexed tuple of matrices*, closely related to the tuples of matrices in [26]. Another notable denotational model built over density operators and quantum operations is [14]. Our typed graphical formalism based on linear logic exploits and expands a recent line of work relating Bayesian networks with proof-nets [11, 8] and with the information flow on type derivations [12].

As discussed in the introduction, the literature on quantum causal models (see *e.g.* [2] and references therein) is vast and multifaceted. We favor the *inference* perspective propounded in [22], a work which is preliminary to QBNs [15]. While we only cite [22], a substantial body of work studies quantum extension of Bayes' rule and Bayesian inference in quantum mechanics and quantum foundations. This is however beyond the scope of our work.

Our goal is a *conservative* extension of Bayesian networks to the quantum setting. An early definition of quantum Bayesian networks—quite different from the one we consider—is given by Tucci [29], where amplitudes replace probabilities. We focus on the work by Henson, Lal, and Pusey [15], that introduces a notion of *generalized Bayesian networks* including the quantum case; recent work in the same line is [17]. A closely connected formalism is that of [13], which focuses on (quantum) non-locality in scenarios with several quantum sources. The results in [15] exploit also the framework of *quantum networks* developed in [4]. It should be noted that *quantum networks* and *quantum Bayesian networks*, despite their similar names, are distinct frameworks with different purposes. In particular, quantum networks cannot encode BNs: a classical operational-probabilistic theory in [4] is not a BN—see the discussion in [15, Section 3.1 and Example 10].

Taking a broader perspective beyond Quantum Bayesian networks, we briefly mention some active research areas which share related aims. The development of a higher-order typed framework for quantum processes [3, 16] is a timely and vibrant research line at the frontier between quantum physics and higher-order computation, exploring quantum operations on quantum operations. Another very active line of research, concerned with compositionality and higher-order structures in both quantum and probabilistic settings, is that of string diagrams, adopting a categorical approach—see *e.g.* [18, 27, 28]. This fruitful line has a perspective different from ours. We noted already that BNs have a dual nature, as tools for *efficient probabilistic inference* and as *causal models*. While we favor the former, the line of developments based on monoidal category theory focuses on causality and is unconcerned with the computational cost of probabilistic reasoning. As observed in [12, Example 9.7], the cost of *actually* computing the semantics of a BN explodes when taking this approach, because a central notion is a product $\otimes$ which behaves like the tensor product of matrices. We give an example in Appendix A.

Our proof-net syntax provides an *exact* characterization of QBNs, setting our work apart from other frameworks. In the last two decades, several papers have developed variants of proof-nets accounting for quantum processes [10, 1, 31, 5, 28]; here we briefly discuss [28] which introduces proof-nets as a tool to analyse *causal* structures. A key ingredient, which is also key in [12] and in our own treatment, is the (polarized) flow of information. The framework in [28] focuses on soundness and completeness (completeness being non-trivial) *w.r.t.* causal string diagrams, building on Retoré's proof-nets: causality is encoded via a non-commutative tensor. In contrast, we present a class of proof-nets that is sound and complete (soundness being non-trivial) *w.r.t.* QBNs. When all data are classical, we recover BNs and [8], where standard inference algorithms (*e.g.* Variable Elimination, Message Passing) are available.

References

1 Samson Abramsky and Ross Duncan. Generalised proof-nets for compact categories with biproducts. *Electronic Notes in Theoretical Computer Science*, 249:3–28, 2009.

2 Jonathan Barrett, Robin Lorenz, and Ognyan Oreshkov. Quantum causal models, 2019. URL: `https://arxiv.org/abs/1906.10726`.

3 Alessandro Bisio and Paolo Perinotti. Theoretical framework for higher-order quantum theory. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 475(2225):20180706, 05 2019. `doi:10.1098/rspa.2018.0706`.

4 Giulio Chiribella, Giacomo Mauro D'Ariano, and Paolo Perinotti. Theoretical framework for quantum networks. *Physical Review A*, 80, August 2009. `doi:10.1103/PhysRevA.80.022339`.

5 Ugo Dal Lago, Claudia Faggian, Benoît Valiron, and Akira Yoshimizu. The geometry of parallelism: classical, probabilistic, and quantum effects. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, Paris, France, January 18-20, 2017*, pages 833–845. ACM, 2017. `doi:10.1145/3009837.3009859`.

6 Adnan Darwiche. Bayesian networks. In *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 467–509. Elsevier, 2008. `doi:10.1016/S1574-6526(07)03011-8`.

7 Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

8 Rémi Di Guardia, Thomas Ehrhard, Jérôme Evrard, and Claudia Faggian. Bayesian networks and proof-nets: the proof-theory of Bayesian inference, 2026. URL: `https://arxiv.org/abs/2602.04045`.

9 Rémi Di Guardia, Thomas Ehrhard, and Claudia Faggian. Quantum Bayesian networks: Compositionality and typing via linear logic, 2026. URL: `https://arxiv.org/abs/2604.26059`.

10 Ross Duncan. *Types for Quantum Computing*. PhD thesis, University of Oxford, 2006.

11 Thomas Ehrhard, Claudia Faggian, and Michele Pagani. The sum-product algorithm for quantitative multiplicative linear logic. In *8th International Conference on Formal Structures for Computation and Deduction, FSCD 2023*, volume 260 of *LIPIcs*, pages 8:1–8:18, 2023. `doi:10.4230/LIPICS.FSCD.2023.8`.

12 Claudia Faggian, Daniele Pautasso, and Gabriele Vanoni. Higher order Bayesian networks, exactly. In *Proceedings of the 51st Annual Symposium on Principles of Programming Languages (POPL)*, volume 8, pages 2514–2546. ACM, January 2024. `doi:10.1145/3632926`.

13 Tobias Fritz. Beyond Bell's theorem: Correlation scenarios. *New Journal of Physics*, 14:103001, 2012. `doi:10.1088/1367-2630/14/10/103001`.

14 Ichiro Hasuo and Naohiko Hoshino. Semantics of higher-order quantum computation via geometry of interaction. *Annals of Pure and Applied Logic*, 168(3):592–637, 2017. `doi:10.1016/J.APAL.2016.10.010`.

15 Joe Henson, Raymond Lal, and Matthew F. Pusey. Theory-independent limits on correlations from generalised Bayesian networks. *New Journal of Physics*, 16(11):113043, 2014. URL: `https://arxiv.org/abs/1405.2572`, `doi:10.1088/1367-2630/16/11/113043`.

16 Timothée Hoffreumon and Ognyan Oreshkov. Projective characterization of higher-order quantum transformations. *Quantum*, 10, January 2026. `doi:10.22331/q-2026-01-21-1978`.

17 Shashaank Khanna, Marina Maciel Ansanelli, Matthew F. Pusey, and Elie Wolfe. Classifying causal structures: Ascertaining when classical correlations are constrained by inequalities. *Physical Review Research*, 6:023038, 2024. `doi:10.1103/PhysRevResearch.6.023038`.

18 Aleks Kissinger and Sander Uijlen. A categorical semantics for causal structure. *Logical Methods in Computer Science*, Volume 15, Issue 3, 2019. URL: `https://lmcs.episciences.org/4426`, `doi:10.23638/LMCS-15(3:15)2019`.

19 Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

20 Olivier Laurent. *Étude de la polarisation en logique*. Thèse de doctorat, Université Aix-Marseille II, 2002.

21 Olivier Laurent. Polarized proof-nets and lambda-μ-calculus. *Theoretical Computer Science*, 290(1):161–188, 2003. `doi:10.1016/S0304-3975(01)00297-3`.

22 Matthew S. Leifer and Robert W. Spekkens. Towards a formulation of quantum theory as a causally neutral theory of Bayesian inference. *Physical Review A*, 88(5):052130, Nov 2013. arXiv:1107.5849. `doi:10.1103/PhysRevA.88.052130`.

23 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2010.

24 Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288, 1986. `doi:10.1016/0004-3702(86)90072-X`.

25 Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK, 2nd edition, 2009.

26 Peter Selinger. Toward a semantics for higher-order quantum computation. In *Proceedings of the 2nd International Workshop on Quantum Programming Languages*, volume 33, pages 127–143. TUCS General Publication, 2004.

27 Will Simmons and Aleks Kissinger. Higher-Order Causal Theories Are Models of BV-Logic. In *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*, volume 241 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 80:1–80:14, 2022. `doi:10.4230/LIPIcs.MFCS.2022.80`.

28 Will Simmons and Aleks Kissinger. A complete logic for causal consistency, 2024. URL: `https://arxiv.org/abs/2403.09297`.

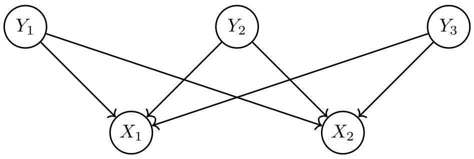
29 Robert R. Tucci. Quantum Bayesian Nets. *International Journal of Modern Physics B*, 9:295–337, 1995. `doi:10.1142/S0217979295000148`.

30 John Watrous. *The Theory of Quantum Information*. Cambridge University Press, Cambridge, 2018.

31 Akira Yoshimizu, Ichiro Hasuo, Claudia Faggian, and Ugo Dal Lago. Measurements in proof nets as higher-order quantum circuits. In *Programming Languages and Systems*, volume 8410 of *Lecture Notes in Computer Science*, pages 258–273. Springer, 2014. ESOP 2014.

A Factors product vs Tensor product

We stress how much the product of factors in the theory of Bayesian networks (noted $\odot$ in our paper) differs from the tensor product of matrices. Consider the following BN (corresponding to the term in [12, Example 9.7]), whose semantics is $\Pr(X_1, X_2, Y_1, Y_2, Y_3)$, a table with 2^5 entries.



The semantics $[\![X_1]\!]$ of the node X_1 is $\Pr(X_1 \mid Y_1, Y_2, Y_3)$ (a factor $\phi_1(x_1, y_1, y_2, y_3)$), which can be seen as a stochastic matrix of size 2^4. Similarly, the semantics $[\![X_2]\!]$ of the node X_2 is $\Pr(X_2 \mid Y_1, Y_2, Y_3)$ and can be seen as a stochastic matrix with 2^4 entries.

By using the respective definitions, it is immediate to check that:

- The tensor product $[\![X_1]\!] \otimes [\![X_2]\!]$ yields a table with 2^8 entries.
- The factors product $[\![X_1]\!] \odot [\![X_2]\!]$ yields a table with 2^5 entries (because non compatible instantiations are never computed).

On the computational cost of the semantical interpretation: $\odot$ vs $\otimes$. *Inference* algorithms on BNs aim at computing the semantics of the model through intermediate, partial computations, *without ever computing the full joint probability* (as unfeasible in practice).

The approaches based on monoidal categories typically focus on *causality*, not on the cost of inference. A central role in the semantics is here played by a product $\otimes$ behaving as the tensor product of matrices. Computing the semantics of n binary random variables easily leads to intermediate computations of size much larger than 2^n, the size of the full joint distribution. Looking again at the BN above, consider the sub-net containing only the nodes X_1 and X_2. Its factor-based semantics is $[\![X_1]\!] \odot [\![X_2]\!]$. In a categorical setting, one computes $[\![X_1]\!] \otimes [\![X_2]\!]$, which is *larger* than the full joint probability distribution (the worst-case scenario in BN's inference). The point is that *inference is not the focus* there.

B Proof of Proposition 18

▶ **Proposition 18.** *Q-factors are* closed *under both product and sum. Moreover:*

1. $(\phi_1 \odot \phi_2) \odot \phi_3 = \phi_1 \odot (\phi_2 \odot \phi_3)$ *provided no qubit appears in the scope of all three Q-factors*
2. $\phi_1 \odot \phi_2 = \phi_2 \odot \phi_1$
3. $\sum_U \sum_V \phi_1 = \sum_V \sum_U \phi_1$
4. $\sum_V (\phi_1 \odot \phi_2) = (\sum_V \phi_1) \odot \phi_2$ *when V is not in the scope of ϕ_2*

Proof. We only prove closeness, the rest being simple computations for each claimed equation.

- Consider $\phi_1 \odot \phi_2$ with ϕ_i a Q-factor over $(\mathbf{X}_i, Q_i)$. We will use the completely positive map $\mathrm{cup}_{\mathcal{L}(\mathcal{H}_Q)} : \mathcal{L}(\mathcal{H}_Q) \otimes \mathcal{L}(\mathcal{H}_Q) \to \mathbb{C}$ defined on the canonical orthonormal basis by

  $$\mathrm{cup}_{\mathcal{L}(\mathcal{H}_Q)}(\mathbf{e} \otimes \mathbf{e}') \stackrel{\text{def}}{=} 1 \text{ if } \mathbf{e} = \mathbf{e}' \text{ and } 0 \text{ otherwise}$$

  and extended to the general case by linearity. Fix $\mathbf{x} \in \mathsf{Val}(\mathbf{X}_1 \cup \mathbf{X}_2)$, with $\mathbf{x}_i$ its restriction to $\mathsf{Val}(\mathbf{X}_i)$. Then, $\phi_1(\mathbf{x}_1)$ is a positive matrix in $\mathcal{L}(\mathcal{H}_{Q_1}) \cong \mathcal{L}(\mathcal{H}_{Q_1 - Q_2}) \otimes \mathcal{L}(\mathcal{H}_{Q_1 \cap Q_2})$ and $\phi_2(\mathbf{x}_2)$ is a positive matrix in $\mathcal{L}(\mathcal{H}_{Q_2}) \cong \mathcal{L}(\mathcal{H}_{Q_1 \cap Q_2}) \otimes \mathcal{L}(\mathcal{H}_{Q_2 - Q_1})$. We want positivity of $(\phi_1 \odot \phi_2)(\mathbf{x}) \in \mathcal{L}(\mathcal{H}_{Q_1 \Delta Q_2}) \cong \mathcal{L}(\mathcal{H}_{Q_1 - Q_2}) \otimes \mathcal{L}(\mathcal{H}_{Q_2 - Q_1})$. We remark that:

  $$(\phi_1 \odot \phi_2)(\mathbf{x}) = \left(\mathrm{Id}_{\mathcal{L}(\mathcal{H}_{Q_1 - Q_2})} \otimes \mathrm{cup}_{\mathcal{L}(\mathcal{H}_{Q_1 \cap Q_2})} \otimes \mathrm{Id}_{\mathcal{L}(\mathcal{H}_{Q_2 - Q_1})}\right)(\phi_1(\mathbf{x}_1) \otimes \phi_2(\mathbf{x}_2))$$

As $\phi_1(\mathbf{x}_1) \otimes \phi_2(\mathbf{x}_2)$ is positive and cup is completely positive, $(\phi_1 \odot \phi_2)(\mathbf{x})$ is positive.

- Consider a Q-factor ϕ over $(\mathbf{Y} \uplus \{X\}, Q)$. Fixing $\mathbf{y} \in \mathsf{Val}(\mathbf{Y})$, as each $\phi(\mathbf{y}, x)$ is a positive matrix, it follows that their sum $(\sum_X \phi)(\mathbf{y}) = \sum_{x \in \mathsf{Val}(X)} \phi(\mathbf{y}, x)$ is positive too.
- Consider a Q-factor ϕ over $(\mathbf{X}, Q \uplus Q')$. For any $\mathbf{x} \in \mathsf{Val}(\mathbf{X})$, $(\sum_{Q'} \phi)(\mathbf{x}) = \mathrm{tr}_{\mathcal{L}(\mathcal{H}_{Q'})}(\phi(\mathbf{x}))$ is positive since $\phi(\mathbf{x})$ is positive and the (partial) trace is completely positive. ◀

C Around Quantum CPTs

Notice that our semantics of quantum Bayesian networks does not distinguish inputs and outputs of nodes, similarly to the factors semantics or to the relational models of Linear Logic. For instance, the interpretation of a node with a quantum input Q_1, a quantum output Q_2 and a classical output X is a function from $\mathsf{Val}(X)$ to positive matrices in $\mathcal{L}(\mathcal{H}_{Q_1} \otimes \mathcal{H}_{Q_2})^+$.

C.1 Intuitions for Quantum CPTs

We give some intuition for Def. 19. It is straightforward to associate a Q-factor to a quantum instrument (we do so in Appendix C.2). The reverse is not always possible, and Q-cpts are those Q-factors corresponding to quantum instruments.

Consider a Q-factor ϕ over $(\{X\} \uplus \mathbf{Y}, Q)$. We want it to be the interpretation of a node X in a quantum Bayesian network: given a density matrix $\rho \in \mathcal{D}(\mathcal{H}_Q)$ and a value $\mathbf{y} \in \mathsf{Val}(\mathbf{Y})$, we want a probability distribution on X. A natural way to do so is by seeing ρ as a Q-factor over $(\emptyset, Q)$ (*i.e.* the function sending the unique element of a singleton to ρ): then, $\phi \odot \rho$ is a Q-factor over $(\{X\} \cup \mathbf{Y}, \emptyset)$, *i.e.* each $(\phi \odot \rho)(x, \mathbf{y})$ is in $\mathbb{R}_0^+$. Therefore, we say that ϕ is a Q-cpt for X given $(\mathbf{Y}, Q)$ if for all $\rho \in \mathcal{D}(\mathcal{H}_Q)$ and $\mathbf{y} \in \mathsf{Val}(\mathbf{Y})$, $\sum_{x \in \mathsf{Val}(X)} (\phi \odot \rho)(x, \mathbf{y}) = 1$.

Similarly, consider a Q-factor ϕ over $(\mathbf{X}, Q \uplus Q')$ that we want to associate to a node Q of a network. Hence, for every density matrix $\rho \in \mathcal{D}(\mathcal{H}_{Q'})$ and value $\mathbf{x} \in \mathsf{Val}(\mathbf{X})$, we should get a density matrix in $\mathcal{D}(\mathcal{H}_Q)$. As before, seeing ρ as a Q-factor over $(\emptyset, Q')$, let us look at the Q-factor $\phi \odot \rho$ over $(\mathbf{X}, Q)$. For ϕ to be a Q-cpt for Q given $(\mathbf{X}, Q')$, we require that for all $\rho \in \mathcal{L}(\mathcal{H}_{Q'})$ and $\mathbf{x} \in \mathsf{Val}(\mathbf{X})$, the positive matrix $(\phi \odot \rho)(\mathbf{x})$ has trace 1.

Def. 19 is simply a reformulation of the two conditions found above.

▶ Remark 40. While complete positivity is relevant for instruments, for Q-factors we only need to consider positive matrices. This is because given $\mathcal{E} : \mathcal{L}(\mathbb{C}) \to \mathcal{L}(\mathcal{H})$—which is isomorphic to a matrix in $\mathcal{L}(\mathcal{H})$—complete positivity collapses to positivity.

C.2 Equivalence between Instrument- and Q-factor-based QBNs

We prove here a claim from Sec. 4: our definition of quantum Bayesian networks with Q-factors is equivalent to the one with quantum instruments from [15], sketched in Sec. 2.3.

▶ **Proposition 41.** *There is a one-to-one correspondence between instrument- and Q-factor-based Bayesian networks, which preserves the semantics.*

As explained in Sec. 4.2, the two syntaxes consider the same underlying DAGs (up to inconsequential details). What we have to do to obtain Prop. 41 is:

- Giving a bijection Ψ between a Q-cpt associated to a node in our presentation, and instruments associated to the same node in [15].
- Proving that composition of instruments corresponds to product of their images by Ψ.

This bijection Ψ is a simple adaptation of the Choi-Jamiolkowski isomorphism to a setting with classical inputs. This builds upon the fact that Q-cpts are defined similarly to Choi states. We first define Ψ, for nodes associated to a r.v. and nodes associated to a register.

▶ **Lemma 42.**

- *Let X be a r.v., $\mathbf{Y}$ a set of r.v.s, and Q a register. There is a bijection between families of quantum instruments $\{\mathcal{E}_{\mathbf{y}} \mid \mathbf{y} \in \mathsf{Val}(\mathbf{Y})\}$ where each $\mathcal{E}_{\mathbf{y}}$ is an instrument $\{\mathcal{E}_{x,\mathbf{y}} \mid x \in \mathsf{Val}(X)\}$ from $\mathcal{L}(\mathcal{H}_{Q'})$ to $\mathbb{C}$, and Q-cpts on X given $(\mathbf{Y}, Q')$.*
- *Let $\mathbf{X}$ be a set of r.v.s, and Q and Q' two registers. There is a bijection between families of quantum operations $\mathcal{E}_{\mathbf{x}}$ from $\mathcal{L}(\mathcal{H}_{Q'})$ to $\mathcal{L}(\mathcal{H}_Q)$, and Q-cpts on Q given $(\mathbf{X}, Q')$.*

Proof. We generalize the definition of Q-cpts (Def. 19) to get both results as particular cases. A Q-factor ϕ over $(\mathbf{X} \uplus \mathbf{Y}, Q \uplus Q')$ is a **Q-cpt for** $(\mathbf{X}, Q)$ **given** $(\mathbf{Y}, Q')$ if $\forall \mathbf{y} \in \mathsf{Val}(\mathbf{Y})$:

$$\left(\sum_{\mathbf{X},Q} \phi\right)(\mathbf{y}) = \mathrm{Id}_{\mathcal{L}(\mathcal{H}_{Q'})}$$

We give a bijection Ψ (along with its inverse Ψ^{-1}) between families of quantum instruments $\{\mathcal{E}_{\mathbf{y}} \mid \mathbf{y} \in \mathsf{Val}(\mathbf{Y})\}$, where each $\mathcal{E}_{\mathbf{y}}$ is a quantum instrument $\{\mathcal{E}_{\mathbf{x},\mathbf{y}} \mid \mathbf{x} \in \mathsf{Val}(\mathbf{X})\}$ from $\mathcal{L}(\mathcal{H}_{Q'})$ to $L(\mathcal{H}_Q)$, and Q-cpts for $(\mathbf{X}, Q)$ given $(\mathbf{Y}, Q')$. To simplify notations, we write $(\mathbf{e}_i)_i$ (resp. $(\mathbf{e}'_k)_k$) the canonical orthonormal basis of $\mathcal{L}(\mathcal{H}_Q)$ (resp. $\mathcal{L}(\mathcal{H}_{Q'})$). We will use the following.

- Call $\mathrm{cup}_{\mathcal{L}(\mathcal{H}_{Q'})} : \mathcal{L}(\mathcal{H}_{Q'}) \otimes \mathcal{L}(\mathcal{H}_{Q'}) \to \mathbb{C}$ the completely positive map defined by

  $$\mathrm{cup}_{\mathcal{L}(\mathcal{H}_{Q'})}(\mathbf{e}'_k \otimes \mathbf{e}'_l) \stackrel{\text{def}}{=} 1 \text{ if } k = l \text{ and } 0 \text{ otherwise}$$

  and extended to the general case by linearity.
- Set $\mathrm{cap}_{\mathcal{L}(\mathcal{H}_{Q'})} \stackrel{\text{def}}{=} \sum_k \mathbf{e}'_k \otimes \mathbf{e}'_k$, which is a positive matrix in $\mathcal{L}(\mathcal{H}_{Q'}) \otimes \mathcal{L}(\mathcal{H}_{Q'})$.

Let us now define Ψ and Ψ^{-1}.

- Given a family of quantum instruments $\{\mathcal{E}_{\mathbf{y}} \mid \mathbf{y} \in \mathsf{Val}(\mathbf{Y})\}$ from $\mathcal{L}(\mathcal{H}_{Q'})$ to $L(\mathcal{H}_Q)$, we set $\Psi(\{\mathcal{E}_{\mathbf{x},\mathbf{y}} \mid \mathbf{x} \in \mathsf{Val}(\mathbf{X})\})$ the Q-factor over $(\mathbf{X} \uplus \mathbf{Y}, Q \uplus Q')$ defined by:

  $$\left(\Psi(\{\mathcal{E}_{\mathbf{x},\mathbf{y}} \mid \mathbf{x} \in \mathsf{Val}(\mathbf{X})\})\right)(\mathbf{x},\mathbf{y}) \stackrel{\text{def}}{=} (\mathrm{Id}_{\mathcal{L}(\mathcal{H}_{Q'})} \otimes \mathcal{E}_{\mathbf{x},\mathbf{y}})(\mathrm{cap}_{\mathcal{L}(\mathcal{H}_{Q'})})$$

- Given a Q-cpt ϕ for $(\mathbf{X}, Q)$ given $(\mathbf{Y}, Q')$, we define for each $\mathbf{y} \in \mathsf{Val}(\mathbf{Y})$ the following set of maps:

  $$\Psi^{-1}_{\mathbf{y}}(\phi) \stackrel{\text{def}}{=} \{\rho \mapsto (\mathrm{cup}_{\mathcal{L}(\mathcal{H}_{Q'})} \otimes \mathrm{Id}_{\mathcal{L}(\mathcal{H}_Q)})(\rho \otimes \phi(\mathbf{x},\mathbf{y})) \mid \mathbf{x} \in \mathsf{Val}(\mathbf{X})\}$$

  and pose $\Psi^{-1}(\phi) \stackrel{\text{def}}{=} \{\Psi^{-1}_{\mathbf{y}}(\phi) \mid \mathbf{y} \in \mathsf{Val}(\mathbf{y})\}$.

By similar computations to that involved for the Choi-Jamiolkowski isomorphism, Ψ sends quantum instruments to Q-cpts, Ψ^{-1} sends Q-cpts to quantum instruments, and Ψ^{-1} is the inverse of Ψ (the later computation using the yanking equations of cup and cap). ◀

Showing the composition of instruments results in the product of their images by Ψ is also a simple computation (using the yanking equations of cup and cap). This proves Prop. 41.

▶ Remark 43. Prop. 23 is a corollary of Prop. 41, using that the semantics of an instrument-based Bayesian network is a probability distribution. Another corollary is that Q-cpts are closed under product, because instruments are closed by composition.