QUANTUM REJECTION SAMPLING FOR NEURO-SYMBOLIC AI

RESEARCH NOTES IN THE ENEXA AND QROM PROJECTS

February 13, 2026

ABSTRACT

We investigate a synergy between explainability in Neuro-Symbolic AI, reflected in tensor network structure, and sparse Quantum Circuits preparing the models. In particular we study a quantum rejection sampling scheme to prepare samples from Computation-Activation Networks. For experiments and prototyping we introduce the python library qcreason.

Contents

1 Introduction

We study inference of a broad model class of Neuro-Symbolic AI by Quantum Computation because:

- Neuro-Symbolic AI can be approached in the tensor network formalism tnreason, unifying logical, neural and probabilistic paradigms of artificial intelligence. Quantum Circuits are also, by the axioms of quantum mechanics, tensor networks. We study the representation of Computation-Activation Networks, the most generic model class in tnreason, by Quantum Circuits.
- Generic tensor network contraction is NP-hard, reflecting the hardness of logical and probabilistic inference. Quantum Computers can prepare states, which measurement distribution are specific network contractions, which can be hard to contract classically. We study schemes to retrieve information about these states, for example by sampling schemes, and use the retrieved information in inference.

We follow two main ideas:

- **Sampling of Computation-Activation Networks:** Prepare quantum states, which measurement statistics can be utilized to prepare samples from Computation-Activation Networks.
- **Quantum Circuits as Contraction providers:** Quantum circuits are contractions of multiple tensors and therefore tensor networks, and measurement probabilities are given by contractions. Here we investigate

how we can exploit these as contraction provider for tnreason. We are inspired by Deutsch-Jozsa algorithm, which we generalize here.

1.1 Quantum Circuits

By its central axioms, quantum mechanics of multiple qubits is formulated by tensors capturing states and discrete time evolutions. Motivated by the structural similarity, we investigate how quantum circuits can be utilized for the tensor-network based approach towards efficient and explainable AI in the tnreason framework.

Potential Advantage: *Quantum Parallelism* (see (Schuld and Petruccione, 2021, Section 3.2.5)).

- Evaluation of multiple function values by single circuit evaluation, we will relate it with the contraction of β^f here.
- Contraction perspective: Loop-tolerant efficient contractions.
- However: Need a generic encoding scheme to exploit this advantage, which is not known yet.
- We here only provide a scheme based on post-selection, which provides a quantum advantage only through amplitude amplification. Without amplitude amplification and post-selection of samples, the encoded distribution is always uniform.
- Further challenge: Application in NISQ devises Preskill (2018), instead of fault-tolerant quantum computers.

Comparison with classical side, which we can call *Tensor Parallelism*: Message-passing schemes for efficient contractions, but exact in limited cases.

Circuit preparing schemes based on approximation:

- Q-Alchemy Araujo et al. (2023), Q-Tucker CITE
- Tensor-Network Optimization based (alternating schemes) Rudolph et al. (2023b,a)

Exact circuit preparing schemes for distributions:

- Grover-Rudolph Grover and Rudolph (2002), but no quantum advantage Herbert (2021)
- Uniform controlled rotations Möttönen et al. (2005)
- Quantum Shannon decomposition Shende et al. (2006)

Circuit simulation: Since qcreason can prepare quantum circuits to arbitrary tensor networks, it can also be used to simulate quantum circuits (with an overhead!).

- Sander et al. (2025b,a)

1.2 Related works

Several applications of Quantum Inference on symbolic AI have been studied. The novelty of our approach is a concise framework extending these approaches and connecting with generic neural decompositions of functions (see computation circuits).

Main approach for sampling: Quantum Inference scheme on Bayesian networks Low et al. (2014)

- Extend to more general tensor networks than Bayesian networks: Computation-Activation Networks

Further literature:

- Schuld and Petruccione (2021): Sect 7.3.2 Reviewing the paper Low et al. (2014) as an application of fault-tolerant quantum computing
- Wittek and Gogolin (2017): Review of Markov Logic Network sampling (which are a special case of Computation-Activation Networks)

Quantum algorithms for linear algebra:

- Contraction by SWAP and Hadamard test (see Appendix)
- Deutsch-Jozsa Algorithm
- HHL algorithm to solve linear equations Harrow et al. (2009)

2 Notation

We present the basic notation of quantum circuits using the tensor network notation used in the tnreason formalism.

2.1 Comparing tensor networks and quantum circuits

First of all, we need to extend to complex tensors, which are maps

$$\tau : \bigtimes_{k\in[d]} [2] \to \mathbb{C}$$

with image in $\mathbb{C}$ instead of $\mathbb{R}$ as in the report.

A coarse comparison of the nomenclature used for quantum circuits and tensor networks:

Quantum Circuit	**Tensor Network**
Qubit	Boolean Variable
Quantum Gate	Unitary Tensor
Quantum Circuit	Tensor Network of unitaries on a directed graph

Some constraints appear for a tensor network to be a quantum circuit

- **Incoming-Outgoing structure:** Variable appear at most once as incoming and at most once as outgoing variables. Those not appearing as outgoing (respectively as incoming) are the input and the output variables of the whole circuit.
- **Unitarity of each gate:** That is the variables of each tensor are bipartite into sets A^{in} and A^{out} of same cardinality and the basis encoding with respect to this bipartition, that is

  $$T_{\text{in}\to\text{out}}[X^{\text{in}}, X^{\text{out}}] : \bigotimes_{k\in A^{\text{in}}} \mathbb{C}^2 \to \bigotimes_{k\in A^{\text{out}}} \mathbb{C}^2 ,$$

  is a unitary map, that is

  $$\left(T_{\text{in}\to\text{out}}\right)^H \circ \left(T_{\text{in}\to\text{out}}\right) = \left\langle T_{\text{in}\to\text{out}}[X^{\text{in}}, Y], \overline{T}_{\text{in}\to\text{out}}[Y, X^{\text{out}}]\right\rangle_{[X^{\text{out}}, X^{\text{in}}]} = \delta\left[X^{\text{out}}, X^{\text{in}}\right].$$

- **Acyclicity:** Incoming and outgoing variables of each tensor core provide a direction of each edge tensor. With respect to this directionality the graph underlying the tensor network has to be acyclic.

The unitary tensors can be aligned layerwise, if and only if the last two assumption hold, i.e. the directed graph is acyclic and each variable appears at most once as an incoming and at most once as an outgoing variable.

2.2 Controlled Single Qubit Gates

We define the rotation gate around the Y-axis by an angle α as

$$R_Y(\alpha)\left[A^{\text{in}}, A^{\text{out}}\right] \coloneqq \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) & -\sin\left(\frac{\alpha}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right) & \cos\left(\frac{\alpha}{2}\right) \end{bmatrix}$$

Further we define the Pauli-X:

$$\sigma_1\left[A^{\text{in}}, A^{\text{out}}\right] \coloneqq \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Controlled single qubit gates are defined using control qubits, where the gate is applied to the target qubit if the control qubits are in a specific state and the identity is applied otherwise. In the tensor network diagrams, we do not distinguish between incoming and outgoing control qubit variables, since the control acts as a Dirac tensor. Thus, controlled unitary with target qubit X_t and control qubits X_c are represented by tensors

$$\mathcal{U}\left[X_{t,\text{in}}, X_{t,\text{out}}, X_c\right]$$

where for each state x_c to the control variables we have that

$$\mathcal{U}\left[X_{t,\text{in}}, X_{t,\text{out}}, X_c = x_c\right]$$

is a unitary matrix acting on the leg space of the target variable.

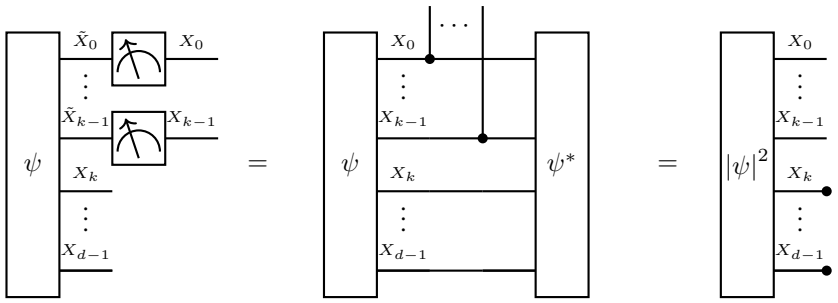


Figure 1: Computational Basis Measurement of a quantum state ψ. The measurement symbols on the left side indicate the measured qubits and the first equation is understood as a definition. In the second equation we sketch, that the measurement distribution is equal to the contraction of the square absolute transform of ψ to the measured variables.

2.3 Measurements and Contractions

2.3.1 Direct measurement

The computational basis measurement of the qubits X_A of a Quantum State $\psi\left[X_{[d]}\right]$ is equal to drawing samples from a distribution

$$\mathbb{P}\left[X_A\right] = \left\langle \psi\left[X_{[d]}\right], \psi^*\left[X_{[d]}\right]\right\rangle_{[X_A]} .$$

Here $\psi^*\left[X_{[d]}\right]$ is the complex conjugate of $\psi\left[X_{[d]}\right]$. When ψ is prepared by a quantum circuit acting on a initial state, the complex conjugate is the hermitean conjugate of the circuit acting on the complex conjgate of the initial state.

We abbreviate these contractions by extending the contraction diagrams with measurement symbols (see Figure 1).

Each complex-valued tensor $\psi\left[X_{[d]}\right]$ has a decomposition into a phase tensor $\phi\left[X_{[d]}\right]$ and an absolute tensor $|\psi|\left[X_{[d]}\right]$ defined by

$$\psi\left[X_{[d]}\right] = \left\langle \exp\left[i \cdot \phi\left[X_{[d]}\right]\right], |\psi|\left[X_{[d]}\right]\right\rangle_{\left[X_{[d]}\right]} .$$

The measurement distribution is depends only on $|\phi|$, that is

$$\mathbb{P}\left[X_{[d]}\right] = \left|\psi\left[X_{[d]}\right]\right|^2 .$$

Note, that this measurement distribution is not sensitive to the phases of the amplitudes.

When only a subset of variables is measured, the distribution is the contraction of the absolute square transform (these operations do not commute)

$$\mathbb{P}\left[X_{\mathcal{U}}\right] = \left\langle \left|\psi\left[X_{[d]}\right]\right|^2 \right\rangle_{[X_{\mathcal{U}}]} .$$

When we are interested in the preparation of quantum states with a specific computational basis measurement distribution, we can restrict to states with vanishing phase cores, that is

$$\psi\left[X_{[d]}\right] = \left\langle \exp\left[i \cdot 0\left[X_{[d]}\right]\right], |\psi|\left[X_{[d]}\right]\right\rangle_{\left[X_{[d]}\right]} = |\psi|\left[X_{[d]}\right].$$

2.3.2 Phase sensitive measurement

When applying a Walsh-Hadamard transform on the qubits $\mathcal{V}\backslash\mathcal{U}$ to be closed (see Figure 2), the probability of measuring the closed qubits in the ground state is

$$\mathbb{P}\left[X_{\mathcal{V}\backslash\mathcal{U}} = 0_{\mathcal{V}\backslash\mathcal{U}}, X_{\mathcal{U}}\right] = \frac{1}{2^{|\mathcal{U}|}} \cdot \left|\left\langle \psi\left[X_{[d]}\right]\right\rangle_{[X_{\mathcal{U}}]}\right|^2 .$$

We thus sample from the contracted quantum state. Note that in this way, the relative phases of the amplitudes are contracted before the absolutes are taken. In this way, the measurement distribution is sensitive to the relative phases. This property is exploited in the Deutsch-Jozsa algorithm, which utilizes signs of the amplitudes.

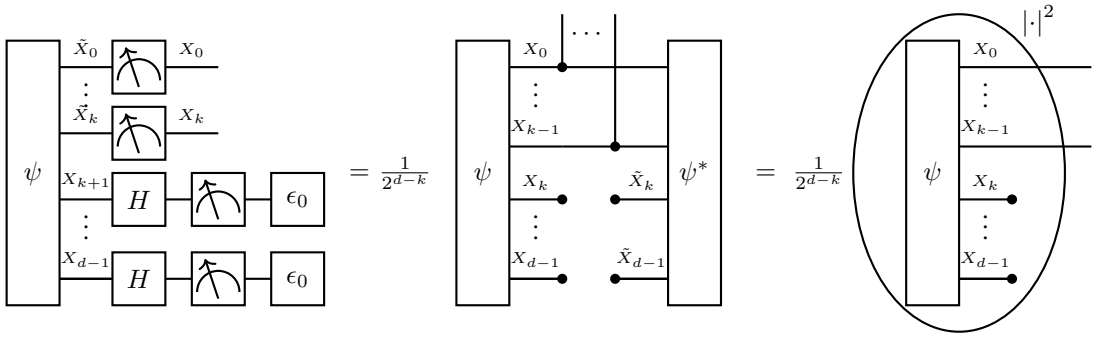


Figure 2: Computational Basis Measurement of a quantum state ψ, after the Walsh-Hadamard Transform on the closed qubits. By the ellipse we indicate a coordinatewise transform to the square of the absolute, after the contraction.

2.3.3 POVM measurements

The main difficulty of using quantum circuits as contraction providers is that we can only extract information through measurements. Therefore measurement is the only way to execute contractions of the circuit, which come with restrictions when interested in contraction with open variables.

The most general measurement formalism is through a POVM, a set $\{E_y : y \in [r]\}$ of positive operators with

$$\sum_{y\in[r]} E_y = I$$

Measuring a pure state $|\psi\rangle$ We then get outcome m with probability

$$\langle\psi|E_y|\psi\rangle$$

We define a measurement variable Y taking indices $y \in [r]$ and a measurement tensor

$$E[Y, X^{\text{in}}, X^{\text{out}}]$$

with slices

$$E[Y = y, X^{\text{in}}, X^{\text{out}}] = E_y\,.$$

Repeating the measurement asymptotically on a state $|\psi\rangle$ prepared by a quantum circuit $\tau^{\mathcal{G}}$ acting on the trivial start state $\mathbb{I}$, we denote the measurement outcome by y^j. In the limit $m \to \infty$ we get almost surely

$$\frac{1}{m}\sum_{j\in[m]} \epsilon_{y^j}\,[Y] \to \left\langle \tau^{\mathcal{G}}[X^{\text{in}}], E[Y, X^{\text{in}}, X^{\text{out}}], \tau^{\tilde{\mathcal{G}}}[X^{\text{out}}]\right\rangle_{[Y]}\,.$$

POVMs to computational basis measurements of subsets of qubits are constructed as products with delta tensors on the non-measured qubits.

3 Graph-Controlled circuits

Let us now introduce the most generic circuit construction schemes used in this work. To this end, we exploit the definition of uniformly controlled unitaries (see Möttönen et al. (2005)), and study their alignment along directed acyclic hypergraphs.

3.1 Definition

We follow the definition of Bayesian networks as tensor networks on directed acyclic hypergraphs (see Goessmann (2025)).

We define graph-controlled circuits using hypergraphs, which nodes are associated with the qubits of the circuit. The hyperedges are tuples of two subsets of the qubits, namely the incoming qubits representing control and the outgoing qubit representing the target of a unitary. To each hyperedge we then associate a uniformly controlled unitary.

Definition 1 (Graph-Controlled Circuit)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed acyclic hypergraph, where each hyperedge has exactly one outgoing node and all nodes appear exactly once as outgoing nodes of an hyperedge. Then a by $\mathcal{G}$ controlled circuit is a decoration of the edges $e = (e^{\text{in}}, \{v\}) \in \mathcal{E}$ by uniformly controlled unitaries*

$$\mathcal{U}^e \left[X_{v,\text{in}}, X_{v,\text{out}}, X_{e^{\text{in}},\text{out}} \right] .$$

Since the hypergraph $\mathcal{G}$ to a graph-controlled circuit is acyclic, we find an order of the qubits, such that to each qubit appearing in the outgoing qubits of a hyperedge is ordered after the incoming qubits. In such way, we have a unique definition of the graph-controlled circuit as a concatenation of the controlled unitaries respecting a directionality order. When there are multiple orders respecting the directionality, then the concatenation respecting the orders all lead to an equivalent circuit.

Example 1 (Generic State Representation)**.** *In Möttönen et al. (2005) a preparation scheme for arbitrary states with real and positive amplitudes by graph-controlled circuits is presented. Therein, the hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed as follows:*

- *Nodes $\mathcal{V} = [d]$*
- *Edges $\mathcal{E} = \{([k], \{k\}) : k \in [d]\}$*

It is possible to realize each controlled unitary by a sequence of single qubit rotations and C-NOTs (see Möttönen et al. (2005)), where the angles are computed using gray codes.

3.2 Activation circuit

A specific scheme to construct graph-controlled circuits with specific Bayesian Networks as measurement distributions are activation circuits. Here we restrict during the preparation procedure to real and positive amplitudes (i.e. prepare q-samples).

Activation circuits are uniformly controlled unitaries (see Möttönen et al. (2005)), where the rotation axis is chosen as the y-axes of the Bloch sphere, and the angles are computed by the function $h(\cdot)$ acting on a function value to be encoded.

We define the angle preparing function on $p \in [0, 1]$ by

$$h(p) = 2 \cdot \cos^{-1}\left(\sqrt{1-p}\right) .$$

For any $p \in [0, 1]$ we then have

$$\left\langle \epsilon_0 \left[A^{\text{in}}\right], R_Y\left(h(p)\right) \left[A^{\text{in}}, A^{\text{out}}\right] \right\rangle_{[A^{\text{out}}]} = \begin{bmatrix} \sqrt{1-p} \\ \sqrt{p} \end{bmatrix} .$$

Definition 2 (Activation circuit)**.** *Given a function*

$$f : \bigtimes_{k \in [d]} [m_k] \to [0, 1]$$

its activation circuit is the uniformly controlled unitary $\mathcal{V}^f \left[A^{\text{in}}, A^{\text{out}}, X_{[d]}\right]$ defined as

$$\mathcal{V}^\tau \left[A^{\text{in}}, A^{\text{out}}, X_{[d]}\right] = \sum_{x_{[d]} \in \bigtimes_{k \in [d]} [m_k]} \epsilon_{x_{[d]}} \left[X_{[d]}\right] \otimes R_Y\left(h(f\left[x_{[d]}\right])\right) \left[A^{\text{in}}, A^{\text{out}}\right] .$$

We will ease our notation by dropping the in and out labels to the control variables. This amounts to understanding the Dirac delta tensors in activation circuits as hyperedges. Along that picture the quantum circuit is a tensor network on hyperedges instead of edges.

Each activation circuit is a graph-controlled circuit, where the corresponding hypergraph consists in $\mathcal{V} = X_{[d]} \cup \{A\}$ and a single edge $\mathcal{E} = (X_{[d]}, \{A\})$.

When we have a probability tensor, its activation circuit be prepared, since all values are in $[0, 1]$. Note that for rejection sampling, only the quotients of the values are important, we can therefore scale the value by a scalar such that the mode is 1.

Tensors $\tau\left[X_{[d]}\right]$ with non-negative coordinates can be encoded after dividing them by their maximum, that is the activation circuit of the function

$$f\left[x_{[d]}\right] = \frac{\tau\left[X_{[d]} = x_{[d]}\right]}{\max_{\tilde{y}_{[p]}} \tau\left[X_{[d]} = \tilde{y}_{[p]}\right]}$$

When the maximum of the tensor is not known, it can be replaced by an upper bound (reducing the acceptance rate of the rejection sampling).

3.2.1 Q-samples

Towards providing additional insights onto the usage of activation circuits we now present them as a scheme to prepare q-samples (see Low et al. (2014)).

In general, we define Q-samples to be quantum states, which measured in the computational basis reproduce a given probability distribution.

Definition 3 (Q-sample)**.** *Given a probability distribution $\mathbb{P} : \bigtimes_{k\in[d]}[2] \to \mathbb{R}$ (i.e. $\langle\mathbb{P}\rangle_{[\varnothing]} = 1$ and $0 \prec \mathbb{P}$) its q-sample is*

$$\psi^{\mathbb{P}}\left[X_{[d]}\right] = \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \sqrt{\mathbb{P}\left[X_{[d]} = x_{[d]}\right]} \cdot \epsilon_{x_{[d]}}\left[X_{[d]}\right] .$$

Q-samples are more restrictive than arbitrary states having a distribution as a measurement distribution, since they demand real and positive amplitudes.

In Low et al. (2014) the Q-sample has been introduced. It prepares a scheme to realize property 1 (purity) + 2 (q-sampling) of a qpdf, but fails to realize property 3 (q-stochasticity).

Example 2 (Q-sample of the uniform distribution)**.** *The q-sample of the uniform distribution can be prepared by Hadamard gates acting on the ground state. This is an example of a Walsh-Hadamard transform (see Sect. B), which can be performed by unary Hadamard gates.*

Generalizing Example 2, Q-samples to independent distributions of boolean variables can be prepared by unary rotations along the y-axis.

3.2.2 Encoding of conditional distributions

Following the schemes in Low et al. (2014), we can prepare the acyclic networks of directed and non-negative tensors by a sequence of controlled rotations. Directed and non-negative tensors correspond with conditional probability distributions and their acyclic networks are Bayesian Networks. We prepare them by activation circuits of functions (see Figure 3)

$$x_{\mathrm{Pa}(k)} \to \mathbb{P}\big[X_k = 1\big|X_{\mathrm{Pa}(k)} = x_{\mathrm{Pa}(k)}\big] .$$

3.2.3 Encoding of Bayesian Networks

We revisit the correspondence of Bayesian Networks with graph-controlled circuits, by showing that any Bayesian Network can be prepared by activation circuits of the conditional probability distributions. Bayesian Networks can be prepared as quantum circuits, where each conditional probability distribution is prepared by an activation circuit.

Theorem 1 (Low et al. (2014))**.** *Let there be a Bayesian Network $\mathbb{P}\left[X_{[d]}\right]$ on a directed acyclic hypergraph $\mathcal{G} = ([d], \mathcal{E})$ with dimensions $m_k = 2$ and decorations of the edges $\mathcal{E} = \{(\mathrm{Pa}(k), \{k\}) : k \in [d]\}$ by conditional probabilities*

$$\mathbb{P}\big[X_k\big|X_{\mathrm{Pa}(k)}\big] .$$

To each $k \in [d]$ we construct the function

$$q_k : \bigtimes_{\tilde{k}\in\mathrm{Pa}(k)} [2] \to [2] \quad , \quad q_k(x_{\mathrm{Pa}(k)}) = \mathbb{P}\big[X_k = 1\big|X_{\mathrm{Pa}(k)} = x_{\mathrm{Pa}(k)}\big] .$$

Then the concatenation of the activation circuits

$$\mathcal{V}^{q_k}\left[X_k^{\mathrm{in}}, X_k, X_{\mathrm{Pa}(k)}\right]$$

respecting the order $[d]$ prepares the Bayesian Network $\mathbb{P}\left[X_{[d]}\right]$ when acting on the initial state $\bigotimes_{k\in[d]} \epsilon_0\left[X_k^{\mathrm{in}}\right]$.

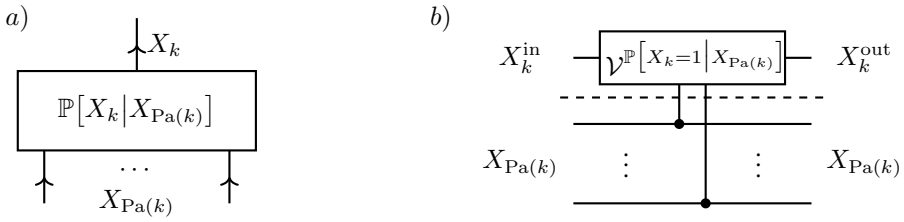


Figure 3: Representation of directed and positive tensor by a controlled rotation. a) Conditional probability tensor $\mathbb{P}\left[X_k \middle| X_{\mathrm{Pa}(k)}\right]$ being a tensor in a Bayesian Network. b) Circuit Encoding as a controlled rotation, which is the Activation circuit of the tensor $\mathbb{P}\left[X_k = 1 \middle| X_{\mathrm{Pa}(k)}\right]$.

Proof. Let $\psi\left[X_{[d]}\right]$ be the state prepared by the activation circuits to the conditional probability distributions. We need to show that the measurement distribution of ψ is the Bayesian Network. For any $x_{[d]} \in \bigtimes_{k\in[d]}[2]$ we have

$$\begin{aligned}\left|\psi\left[X_{[d]} = x_{[d]}\right]\right|^2 &= \prod_{k\in[d]} \left(\mathcal{V}^{q_k}\left[X_k^{\mathrm{in}} = 0, X_k = x_k, X_{\mathrm{Pa}(k)} = x_{\mathrm{Pa}(k)}\right]\right)^2 \\ &= \prod_{k\in[d]} \mathbb{P}\left[X_k = x_k \middle| X_{\mathrm{Pa}(k)} = x_{\mathrm{Pa}(k)}\right] \\ &= \mathbb{P}\left[X_{[d]} = x_{[d]}\right] .\end{aligned}$$

Alternatively, we can proof the statement by induction using the conditional probability encoding lemma. □

Example 3 (Encoding of discretized distributions)**.** *We here review the preparation of a discretized distribution from Grover and Rudolph (2002) as an application of activation circuits. Let $\mathbb{Q}[X]$ be a continuous probability distribution of a variable X taking values in $[0,1]$. For any $d \in \mathbb{N}$ we define its discretization of order d as a discrete probability distribution $\mathbb{P}\left[X_{[d]}\right]$ with coordinates*

$$\mathbb{P}^d\left[X_{[d]} = x_{[d]}\right] = \mathbb{Q}\left[X \in [\sum_{k\in[d]} 2^{-(k+1)}x_k, 2^{-d} + \sum_{k\in[d]} 2^{-(k+1)}x_k]\right] .$$

Then we have for any $p \in \mathbb{N}$ with $p \leq d$

$$\mathbb{P}^p\left[X_{[p]}\right] = \left\langle \mathbb{P}\left[X_{[d]}\right]\right\rangle_{\left[X_{[p]}\right]} .$$

The discretization order is thus increased by the contraction

$$\mathbb{P}^{d+1}\left[X_{[d+1]}\right] = \left\langle \mathbb{P}^d\left[X_{[d]}\right], \mathbb{P}^{d+1}\left[X_d | X_{[d]}\right]\right\rangle_{\left[X_{[d+1]}\right]} .$$

Having an activation circuit preparing $\mathbb{P}^d$ we can therefore prepare $\mathbb{P}^{d+1}$ by adding the activation circuit for the function

$$x_{[d]} \to \mathbb{P}^{d+1}\left[X_d = 1 | X_{[d]} = x_{[d]}\right] .$$

Note that the function evaluation amounts to the integration of $\mathbb{Q}$ as

$$\mathbb{P}^{d+1}\left[X_d = 1 | X_{[d]} = x_{[d]}\right] = \frac{\mathbb{Q}\left[X \in [2^{-(d+1)} + \sum_{k\in[d]} 2^{-(k+1)}x_k, 2^{-d} + \sum_{k\in[d]} 2^{-(k+1)}x_k]\right]}{\mathbb{Q}\left[X \in [\sum_{k\in[d]} 2^{-(k+1)}x_k, 2^{-d} + \sum_{k\in[d]} 2^{-(k+1)}x_k]\right]} .$$

3.3 Equivalence with Bayesian Networks

Above we have shown, how Bayesian Networks can be prepared by activation circuits, which are a class of graph-controlled circuits. We now show, that the measurement distribution of any graph-controlled circuit is a Bayesian Network. It then follows that the set of distributions prepared by graph-controlled circuits is exactly the set of Bayesian Networks.

This is closely connected to the chain decomposition of probability distributions, and their construction. In Low et al. (2014) Bayesian Networks are prepared by graph-controlled circuits. When in the graph $\mathcal{G}$ each node appears at most once as outgoing node, it is also the hypergraph to a family of Bayesian Networks. The measurement distributions of a state prepared by a $\mathcal{G}$-controlled Circuit acting on disentangled initial states are exactly the Bayesian Networks with respect to $\mathcal{G}$.

Theorem 2. *Let $\mathcal{G}$ be a directed acyclic hypergraph, such that node appears at most once as an outgoing node. The measurement distributions of the by $\mathcal{G}$ controlled circuits acting on disentangled initial states are equal to the Bayesian Networks on $\mathcal{G}$.*

Lemma 1. *Let $\mathcal{G}$ be a directed acyclic hypergraph, such that node appears at most once as an outgoing node. Then any Bayesian network on $\mathcal{G}$ can be prepared by a $\mathcal{G}$-controlled circuit with activation circuits of the conditional probability tensors.*

Proof. Let $\mathbb{P}\left[X_{[d]}\right]$ be a Bayesian network on the graph $\mathcal{G}$. Enumerate the nodes $\mathcal{V}$ of the $\mathcal{G}$ by $[d]$, such that for each $k \in [d]$ we have $\mathrm{Pa}(k) \subset [k]$. Then define a $\mathcal{G}$-controlled circuit, by choosing for each $k \in [d]$ controlled unitaries which satisfy

$$\mathcal{U}^k\left[X_k^{\mathrm{in}} = 0, X_{v,\mathrm{out}}, X_{\mathrm{Pa}(k),\mathrm{out}}\right] = \sqrt{\mathbb{P}\big[X_k \big| X_{\mathrm{Pa}(k)}\big]}\,.$$

Here we specified only the action of the controlled unitary on the basis vector $\epsilon_0\left[X_k\right]$, the action on $\epsilon_1\left[X_k\right]$ can be chosen by an arbitrary orthogonal unit vector. For more explicit construction, see the activation circuits. Any such defined $\mathcal{G}$-controlled circuit acting on the initial state $\bigotimes_{k\in[d]} \epsilon_0\left[X_k\right]$ prepares a quantum state $\psi\left[X_{[d]}\right]$ with measurement distribution

$$\left|\psi\right|^2\left[X_{[d]}\right]\,.$$

Given arbitrary $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ we have

$$\left|\psi\right|^2\left[X_{[d]} = x_{[d]}\right] = \prod_{k\in[d]} \mathbb{P}\big[X_k = x_k \big| X_{\mathrm{Pa}(k)} = x_{\mathrm{Pa}(k)}\big] = \mathbb{P}\left[X_{[d]} = x_{[d]}\right]\,.$$

Here we used in the last equation, that $\mathbb{P}\left[X_{[d]}\right]$ is a Bayesian network. Since the equivalence holds for any coordinate, this establishes the equivalence of the measurement distribution of $\psi\left[X_{[d]}\right]$ and $\mathbb{P}\left[X_{[d]}\right]$. □

While we have already shown by Lem. 1 that arbitrary Bayesian Networks can be prepared by graph-controlled circuits, we now show the converse, namely that any distribution prepared by graph-controlled circuits is a Bayesian Network. To this end, we use the characterization of Bayesian Networks by the conditional independencies they encode (see Koller and Friedman (2009)). To be more precise, we need to show that any node variable is conditionally independent of its non-descendants given its parents.

Lemma 2. *Let $(\mathcal{G},\mathcal{U})$ be a $\mathcal{G}$-controlled circuit acting on a disentangled initial state and $\mathbb{P}\left[X_{\mathcal{V}}\right]$ the corresponding measurement distribution. Then we have for each $v \in \mathcal{V}$ the conditional independence*

$$\left(X_v \perp X_{\mathrm{NonDes}(v)}\right) \,\middle|\, X_{\mathrm{Pa}(v)}\,.$$

Proof. We choose to a given $v \in \mathcal{V}$ an enumeration $[d]$ of the nodes, such that for each $k \in [d]$ we have $\mathrm{Pa}(k) \subset [k]$ and for the enumerator $\tilde{k}$ of v we further have $\mathrm{NonDes}(\tilde{k}) \subset [\tilde{k}]$. Let $\mathbb{P}\left[X_{[d]}\right]$ be the measurement distribution of the $\mathcal{G}$-controlled circuit acting on a disentangled initial state $\bigotimes_{k\in[d]} \psi^k\left[X_k\right]$ and choose arbitrary $x_{[k]}$. We then have

$$\begin{aligned}
\mathbb{P}\left[X_{\tilde{k}}, X_{[k]} = x_{[k]}\right] &= \left\langle \left(\bigcup_{k\in[d]} \{\mathcal{U}^k, \mathcal{U}^{k,\dagger}, \psi^k, \psi^{k,*}\right) \} \cup \left(\bigcup_{k\in[\tilde{k}]} \epsilon_{x_k}\left[X_k^{\mathrm{out}}\right] \right) \right\rangle_{\left[X_{\tilde{k}}\right]} \\
&= \left\langle \bigcup_{k\in[\tilde{k}]} \{\mathcal{U}^k, \mathcal{U}^{k,\dagger}, \psi^k, \psi^{k,*}, \epsilon_{x_k}\left[X_k^{\mathrm{out}}\right]\} \right\rangle_{\left[X_{\tilde{k}}\right]} \\
&= \left| \left\langle \mathcal{U}^{\tilde{k}}\left[X_{\tilde{k},\mathrm{in}}, X_{\tilde{k},\mathrm{out}}\right], X_{\mathrm{Pa}(\tilde{k}),\mathrm{out}} = x_{\mathrm{Pa}(\tilde{k}),\mathrm{out}} \right\rangle_{\left[X_{\tilde{k},\mathrm{out}}\right]} \right|^2 \\
&\quad \cdot \prod_{k\in[\tilde{k}]} \left(\left\langle \psi^k\left[X_k^{\mathrm{in}}\right], \mathcal{U}^k\left[X_k^{\mathrm{in}}, X_k^{\mathrm{out}}, X_{\mathrm{Pa}(k),\mathrm{out}} = x_{\mathrm{Pa}(k),\mathrm{out}}\right] \right\rangle_{[\varnothing]} \right)^2
\end{aligned}$$

Here we used in the second equation the unitarity of the controlled unitaries to $k \notin [\tilde{k}]$. Since the indices $x_{[\tilde{k}]/\mathrm{Pa}(\tilde{k})} = x_{\mathrm{NonDes}(\tilde{k})}$ appear only in the constant term, we conclude

$$\mathbb{P}\left[X_{\tilde{k}} \middle| X_{[k]}\right] = \mathbb{P}\left[X_{\tilde{k}} \middle| X_{\mathrm{Pa}(\tilde{k})}\right] \otimes \mathbb{I}\left[X_{\mathrm{NonDes}(\tilde{k})}\right] ,$$

which establishes the conditional independence $\left(X_v \perp X_{\mathrm{NonDes}(v)}\right) \mid X_{\mathrm{Pa}(v)}$. □

Proof of Thm. 2. The theorem follows directly from Lem. 1 and Lem. 2, using that Bayesian Networks are characterized by the conditional independence of each variable to its non-descendants given its parents. □

Example 4 (Chain decomposition)**.** *The hypergraph in Example 1 corresponds with a family of Bayesian Networks subsuming any probability distribution. This can be verified based on the chain decomposition of a generic distribution.*

Another question is, whether each quantum state, which measurement distribution is a Bayesian Network can be prepared by a $\mathcal{G}$-controlled circuit. This is not always the case, since the phase tensor does not influence the measurement distribution. Any phase tensor, of a by $\mathcal{G}$-controlled circuit prepared state has however a decomposition

$$\phi\left[X_{[d]}\right] = \sum_{k \in [d]} \phi^k \left[X_k, X_{\mathrm{Pa}(k)}\right] \otimes \mathbb{I}\left[X_{[d]/\{\{k\} \cup \mathrm{Pa}(k)\}}\right] ,$$

where the phase cores ϕ^k can be read of the controlled unitaries. When there are phase cores which do not have such a decomposition, the corresponding states are not representable. Möttönen et al. (2005) shows, that when the graph is chosen as in Example 1, then also the phases can be represented.

3.4 Ancilla Augmentation

Graph controlled circuits prepare Bayesian Networks, and do not apply to undirected graphical models such as Markov Networks and Computation-Activation Networks. We therefore introduce in the following ancilla augmentation, which is a method to represent Markov Networks as conditioned Bayesian Network. This enables usage of the previously used preparation schemes for Bayesian Networks.

3.4.1 Post-selection by Ancilla Variables

First of all we define ancilla augmentation and then show how they can be prepared as measurement distributions using activation circuits.

Definition 4 (Ancilla Augmentation of a Distribution)**.** *Let $\mathbb{P}\left[X_{[d]}\right]$ be a probability distribution of variables $X_{[d]}$. Another distribution $\tilde{\mathbb{P}}[X_{[d]}, A_{[p]}]$ of variables $X_{[d]}$ and ancilla variables $A_{[p]}$ is called an ancilla augmentation of $\mathbb{P}\left[X_{[d]}\right]$, if there is $\lambda > 0$ with*

$$\lambda \cdot \mathbb{P}\left[X_{[d]}\right] = \tilde{\mathbb{P}}[X_{[d]}, A_{[p]} = 1_{[p]}] .$$

We refer to λ as the corresponding acceptance probability.

Note that $\lambda \leq 1$ since

$$\lambda = \left\langle \lambda \cdot \mathbb{P}\left[X_{[d]}\right] \right\rangle_{[\varnothing]} = \left\langle \tilde{\mathbb{P}}[X_{[d]}, A_{[p]} = 1_{[p]}] \right\rangle_{[\varnothing]} \leq \left\langle \tilde{\mathbb{P}}[X_{[d]}, A_{[p]}] \right\rangle_{[\varnothing]} = 1 .$$

Sampling from the distribution can be done by rejection sampling, also called post selection, on the ancilla augmented distribution in the following way. When drawing a sample $(x_{[d]}, a)$ from $\tilde{\mathbb{P}}[X_{[d]}, A]$ and rejecting it whenever $a = 0$, the accepted samples are distributed as $\mathbb{P}\left[X_{[d]}\right]$. We then have λ as the probability of accepting a sampling in this rejection sampling method. The acceptance probability is critical in determining the efficiency of this approach, since we need to draw $\sim \frac{1}{\lambda}$ samples from $\tilde{\mathbb{P}}[X_{[d]}, A]$ to get an accepted sample.

Since

$$\tilde{\mathbb{P}}[X_{[d]}, A] = \left\langle \tilde{\mathbb{P}}[X_{[d]}], \tilde{\mathbb{P}}[A|X_{[d]}] \right\rangle_{\left[X_{[d]}, A\right]} \tag{1}$$

the described sampling procedure is equal to drawing $x_{[d]}$ from the marginal

$$\tilde{\mathbb{P}}[X_{[d]}]$$

and accepting the sample with probability

$$\tilde{\mathbb{P}}[A = 1|X_{[d]}] .$$

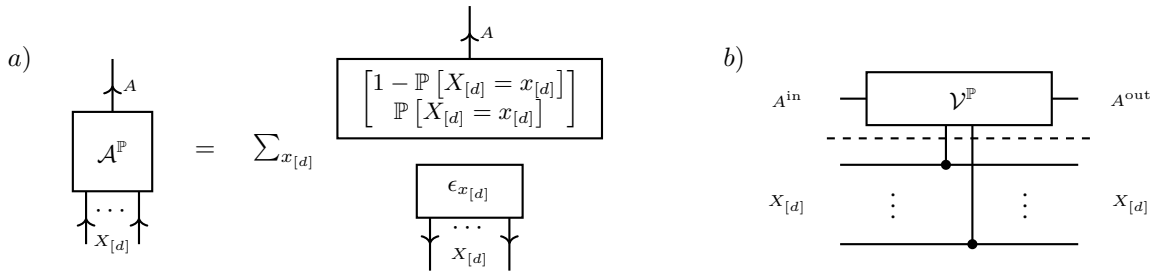


Figure 4: Ancilla augmentation of a distribution $\mathbb{P}\left[X_{[d]}\right]$. a) Augmented distribution $\mathcal{A}^{\mathbb{P}}\left[A, X_{[d]}\right]$ with the property that $\mathbb{P}\left[X_{[d]}\right] = \mathcal{A}^{\mathbb{P}}\left[X_{[d]}|A=1\right]$. b) Preparation of the augmented distribution by the activation circuit of $\mathbb{P}\left[X_{[d]}\right]$.

3.4.2 Construction based on Uniform Distributions

We construct ancilla augmentations based on the decomposition (1) where the marginal of $X_{[d]}$ is the uniform distribution, that is

$$\tilde{\mathbb{P}}[X_{[d]}] = \frac{1}{\prod_{k\in[d]} m_k}\mathbb{I}\left[X_{[d]}\right]$$

and we construct the conditional probability of the ancilla variable such that

$$\tilde{\mathbb{P}}[A=1|X_{[d]}] = \lambda \cdot \left(\prod_{k\in[d]} m_k\right)\mathbb{P}\left[X_{[d]}\right] .$$

For the latter we can apply the preparation schemes of conditioned distributions, which have been studied above for Bayesian Networks.

We denote for any non-negative tensor $\tau\left[X_{[d]}\right]$ we construct

$$\mathcal{A}^{\tau,\lambda}\left[A, X_{[d]}\right] = \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \epsilon_{x_{[d]}}\left[X_{[d]}\right] \otimes \Big(\lambda\cdot\big(\prod_{k\in[d]} m_k\big)\cdot\tau\left[X_{[d]}=x_{[d]}\right]\cdot\epsilon_1\left[A\right] + \big(1-\lambda\cdot\big(\prod_{k\in[d]} m_k\big)\cdot\tau\left[X_{[d]}=x_{[d]}\right]\big)\cdot\epsilon_0\left[A\right]\Big).$$

For this to be a conditional distribution we need to demand that

$$\max_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \lambda\cdot\big(\prod_{k\in[d]} m_k\big)\cdot\tau\left[X_{[d]}=x_{[d]}\right] \leq 1\,,$$

which is equivalent to

$$\lambda \leq \left(\prod_{k\in[d]} m_k\right)^{-1}\cdot\left(\max_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]}\tau\left[X_{[d]}=x_{[d]}\right]\right)^{-1} .$$

Note that if $\tau\left[X_{[d]}\right]$ is the uniform distribution $\mathbb{I}\left[X_{[d]}|\varnothing\right]$, then we can achieve the best acceptance rate $\lambda = 1$.

Given a distribution $\mathbb{P}\left[X_{[d]}\right]$ we add an ancilla variable A and construct the augmented distribution (see Figure 4)

Lemma 3. *The activation circuit to a probability distribution $\mathbb{P}\left[X_{[d]}\right]$ applied on the initial state*

$$\sqrt{\frac{1}{\prod_{k\in[d]} m_k}}\mathbb{I}\left[X_{[d]}\right]\otimes\epsilon_0\left[A\right]$$

prepares a quantum state ψ, which is the Q-sample of the ancilla augmentation

$$\left\langle \mathcal{A}^{\mathbb{P},\frac{1}{m}}\left[A, X_{[d]}\right]\right\rangle_{\left[A,X_{[d]}|\varnothing\right]}$$

with the acceptance probability $m^{-1} \coloneqq \left(\prod_{k\in[d]m_k}\right)^{-1}$.

Proof. We first show, that $\left\langle \mathcal{A}^{\mathbb{P},\frac{1}{m}}\left[A, X_{[d]}\right]\right\rangle_{\left[A,X_{[d]}|\varnothing\right]}$ is indeed an ancilla augmentation. We have

$$\lambda = \left\langle \mathcal{A}^{\mathbb{P}}\left[A, X_{[d]}\right]\right\rangle_{[A=1]} = \frac{1}{m}\sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \mathbb{P}\left[X_{[d]} = x_{[d]}\right] = \frac{1}{m}\,.$$

and for any $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ it holds that

$$\left\langle \mathcal{A}^{\mathbb{P},\frac{1}{m}}\left[A, X_{[d]}\right]\right\rangle_{\left[A=1,X_{[d]}=x_{[d]}|\varnothing\right]} = \frac{1}{\lambda}\frac{1}{\prod_{k\in[d]} m_k}\mathbb{P}\left[X_{[d]} = x_{[d]}\right] = \mathbb{P}\left[X_{[d]} = x_{[d]}\right]\,,$$

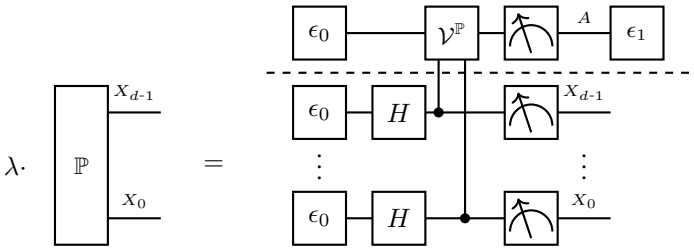
The tensor $\left\langle \mathcal{A}^{\mathbb{P},\frac{1}{m}}\left[A, X_{[d]}\right]\right\rangle_{\left[A,X_{[d]}|\varnothing\right]}$ is thus an ancilla augmentation for $\mathbb{P}$.

We continue to show that the quantum state ψ prepared by the activation circuit is the Q-sample of $\left\langle \mathcal{A}^{\mathbb{P},\frac{1}{m}}\left[A, X_{[d]}\right]\right\rangle_{\left[A,X_{[d]}|\varnothing\right]}$. Then we have for any $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$

$$\psi\left[A, X_{[d]} = x_{[d]}\right] = \sqrt{\frac{1}{\prod_{k\in[d]} m_k}}\left(\sqrt{\mathbb{P}\left[X_{[d]} = x_{[d]}\right]}\cdot\epsilon_1\left[A\right] + \sqrt{(1-\mathbb{P}\left[X_{[d]} = x_{[d]}\right])}\cdot\epsilon_0\left[A\right]\right),$$

which is indeed the Q-sample. □

In our tensor network circuit notation, Lem. 3 states the equation



for the acceptance probability (see proof of Lem. 3)

$$\lambda = \frac{1}{\prod_{k\in[d]} m_k}\,.$$

Note that the acceptance probability is exponentially small in the number of variables. If $\mathbb{P}\left[X_{[d]}\right]$ is not uniform, we can improve by building the activation circuit to the function

$$\frac{1}{\max_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]}\mathbb{P}\left[X_{[d]} = x_{[d]}\right]}\cdot\mathbb{P}\left[X_{[d]}\right]\,.$$

3.4.3 Best Acceptance Probability by ∞-Renyi Divergence

Let us now show, how the best acceptance probability in sets of ancilla augmented distributions is connected with ∞-Renyi divergences. For $0 < \alpha < \infty$ and $\alpha \neq 1$ the Renyi Divergence is defined as

$$D_\alpha\left[\mathbb{P}||\mathbb{Q}\right] = \frac{1}{1-\alpha}\ln\left[\sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]}\frac{\mathbb{P}\left[X_{[d]} = x_{[d]}\right]^\alpha}{\mathbb{Q}\left[X_{[d]} = x_{[d]}\right]^{\alpha-1}}\right]\,.$$

The ∞ Renyi Divergence is the limit $\alpha \to \infty$

$$D_\infty\left[\mathbb{P}||\mathbb{Q}\right] = \ln\left[\max_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]}\frac{\mathbb{P}\left[X_{[d]} = x_{[d]}\right]}{\mathbb{Q}\left[X_{[d]} = x_{[d]}\right]}\right]\,.$$

Theorem 3. *Let $\mathbb{P}\left[X_{[d]}\right]$ and $\mathbb{Q}[X_{[d]}]$ be distributions, such that for each $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ with $\mathbb{P}\left[X_{[d]} = x_{[d]}\right] > 0$ we also have $\mathbb{Q}[X_{[d]} = x_{[d]}] > 0$. Let us consider all ancilla augmentations $\tilde{\mathbb{P}}[X_{[d]}, A]$ which marginal is $\mathbb{Q}[X_{[d]}]$, that is*

$$\mathbb{Q}[X_{[d]}] = \left\langle \tilde{\mathbb{P}}[X_{[d]}, A] \right\rangle_{\left[X_{[d]}\right]} .$$

The best acceptance probability among these ancilla augmentations is then

$$\lambda = \exp\left[-D_\infty\left[\mathbb{P}||\mathbb{Q}\right]\right] ,$$

and achieved when for any $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$

$$\tilde{\mathbb{P}}[A = 1|X_{[d]} = x_{[d]}] = \frac{\mathbb{P}\left[X_{[d]} = x_{[d]}\right]}{\mathbb{Q}\left[X_{[d]} = x_{[d]}\right]} \cdot \left(\max_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \frac{\mathbb{P}\left[X_{[d]} = x_{[d]}\right]}{\mathbb{Q}\left[X_{[d]} = x_{[d]}\right]} \right)^{-1} .$$

Proof. Any $\tilde{\mathbb{P}}[X_{[d]}, A]$ with marginal $\mathbb{Q}\left[X_{[d]}\right]$ has a decomposition

$$\tilde{\mathbb{P}}[X_{[d]}, A] = \left\langle \mathbb{Q}\left[X_{[d]}\right], \tilde{\mathbb{P}}[A|X_{[d]}] \right\rangle_{\left[X_{[d]}, A\right]}$$

and in order to be an ancilla augmentation for $\mathbb{P}\left[X_{[d]}\right]$ we find $\lambda > 0$ with

$$\tilde{\mathbb{P}}[X_{[d]}, A = 1] = \lambda \cdot \mathbb{P}\left[X_{[d]}\right] .$$

It follows that for each $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$

$$1 \geq \tilde{\mathbb{P}}[A = 1|X_{[d]} = x_{[d]}] = \lambda \cdot \frac{\mathbb{P}\left[X_{[d]} = x_{[d]}\right]}{\mathbb{Q}\left[X_{[d]} = x_{[d]}\right]} .$$

and thus

$$\lambda \leq \left(\max_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \frac{\mathbb{P}\left[X_{[d]} = x_{[d]}\right]}{\mathbb{Q}\left[X_{[d]} = x_{[d]}\right]} \right)^{-1} . \tag{2}$$

For the in the claim constructed ancilla augmentation, we have the acceptance probability

$$\begin{aligned}
\tilde{\mathbb{P}}[A = 1] &= \mathbb{E}_{x_{[d]}\sim\mathbb{Q}} \left[\frac{\mathbb{P}\left[X_{[d]} = x_{[d]}\right]}{\mathbb{Q}\left[X_{[d]} = x_{[d]}\right]} \cdot \left(\max_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \frac{\mathbb{P}\left[X_{[d]} = x_{[d]}\right]}{\mathbb{Q}\left[X_{[d]} = x_{[d]}\right]} \right)^{-1} \right] \\
&= \left\langle \mathbb{Q}\left[X_{[d]}\right], \left(\mathbb{Q}\left[X_{[d]}\right]\right)^{-1}, \mathbb{P}\left[X_{[d]}\right] \right\rangle_{[\varnothing]} \left(\max_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \frac{\mathbb{P}\left[X_{[d]} = x_{[d]}\right]}{\mathbb{Q}\left[X_{[d]} = x_{[d]}\right]} \right)^{-1} \\
&= \left(\max_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \frac{\mathbb{P}\left[X_{[d]} = x_{[d]}\right]}{\mathbb{Q}\left[X_{[d]} = x_{[d]}\right]} \right)^{-1} \\
&= \exp\left[-D_\infty\left[\mathbb{P}||\mathbb{Q}\right]\right] .
\end{aligned}$$

The constructed ancilla augmented distribution satisfies the bound (2) on the acceptance probability holds straight, and it is thus the one with largest acceptance rate. □

Note, that in the section above we have constructed ancilla augmentations with uniform $\mathbb{Q}$ by activation circuits. For these the optimal acceptance rate is

$$\lambda = \left(\prod_{k\in[d]} m_k \right)^{-1} \cdot \left(\max_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \mathbb{P}\left[X_{[d]} = x_{[d]}\right] \right)^{-1} .$$

Thus, $\lambda = 1$ is only achievable if $\mathbb{P}\left[X_{[d]}\right]$ is itself the uniform distribution.

Extension: We could increase the acceptance probability, when we sample from a proposal distribution $\mathbb{Q}$ with smaller ∞ Renyi divergence to $\mathbb{P}$. When sampling with Quantum Circuits, this could be implemented by a state preparation for the distributed variables, before the Computation Activation Circuit is applied. It would be interesting to train variational quantum circuits for this task. However, when we want to apply the same scheme as above, one needs to encode $\mathbb{Q}$ into the ancilla preparing rotations, so $\mathbb{Q}$ would need to be an elementary Computation-Activation Network as well.

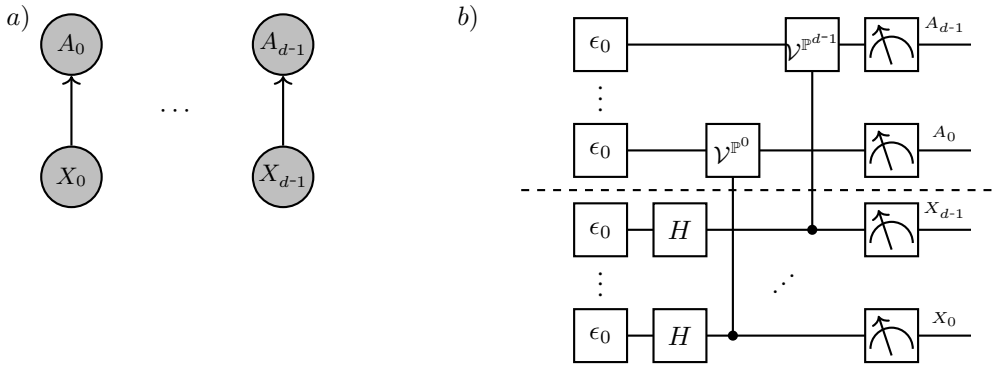


Figure 5: Ancilla augmentation of an elementary distribution (which variables are pairwise independent). a) Hypergraph to the Bayesian Network representing the ancilla augmentation. b) Circuit preparing the ancilla augmented distribution by activation circuits acting on the disentangled initial state.

3.4.4 Augmentation of Markov Networks

Let us now show how multiple ancilla variables can be designed to sample from arbitrary Markov Networks.

Example 5 (Elementary Tensors)**.** *Let there be an elementary distribution*

$$\mathbb{P}\left[X_{[d]}\right] = \bigotimes_{k\in[d]} \mathbb{P}^k\left[X_k\right] ,$$

that is, all variables are pairwise independent. Then the ancilla augmentation of the distribution is given by the Bayesian Network with the hypergraph shown in Figure 5a). Compared with the ancilla augmentation of a generic distribution this augmentation is sparser in the circuit depth (see Figure 5b), but introduces multiple ancilla variables. The sparsity originates since each ancilla variable is controlled by a single qubit. The total amount of controlled rotations is thus $\sum_{k\in[d]}[m_k]$ *instead of* $\prod_{k\in[d]}[m_k]$ *for the generic implementation.*

Let there be a Markov Network, i.e. any non-vanishing tensor network of non-negative tensors, which is understood as a probability disribution after normalization (see Goessmann et al. (2026)). For an example of a Markov Network see Figure 6a.

We apply a similar ancilla augmentation on any hypercore of a Markov Network, as has been done above for probability distributions. To prepare, we need to scale each core by dividing by its maximum (or any upper bound on its maximum), such that any coordinate is smaller than 1. We introduce for each core an boolean ancilla variable A_e (while X_e is the tuple of distributed variables collected in the hyperedge e) then build directed tensors

$$\begin{aligned}\mathcal{A}^{\tau^e,\lambda^e}\left[A_e, X_e\right] = \sum_{x_e} \epsilon_{x_e}\left[X_e\right] \otimes \Big(&\lambda^e \cdot \big(\prod_{k\in e} m_k\big) \cdot \tau^e\left[X_e = x_e\right] \cdot \epsilon_1\left[A\right] \\ &+ \big(1 - \lambda^e \cdot \big(\prod_{k\in e} m_k\big) \cdot \tau^e\left[X_e = x_e\right]\big) \cdot \epsilon_0\left[A\right]\Big) .\end{aligned}$$

where we choose the maximal values

$$\lambda^e = \left(\prod_{k\in e} m_k\right)^{-1} \cdot \left(\max_{x_e} \tau^e\left[X_e = x_e\right]\right)^{-1} ,$$

such that $\mathcal{A}^e$ stays non-negative and can be interpreted as a conditional probability distribution. Note, that λ^e is not an acceptance rate itself, since τ^e is not always a distribution.

Lemma 4. *When augmenting all cores of a Markov Network with distinct ancilla variables in this way (see Figure 6b), we get an ancilla augmentation of the Markov Network.*

$$\tilde{\mathbb{P}}[A_{\mathcal{E}}, X_{[d]}] = \left\langle \{\mathcal{A}^{\tau^e,\lambda^e}\left[A_e, X_e\right] \,:\, e \in \mathcal{E}\} \cup \{\mathbb{I}\left[X_{[d]}|\varnothing\right]\}\right\rangle_{\left[A_{\mathcal{E}}, X_{[d]}\right]} .$$

Proof. $\tilde{\mathbb{P}}[A_{\mathcal{E}}, X_{[d]}]$ is a Bayesian Network (see Figure 6d) and thus a distribution. We further have

$$\begin{aligned}\tilde{\mathbb{P}}[A_{\mathcal{E}} = 1_{\mathcal{E}}, X_{[d]}] &= \left\langle \{\mathcal{A}^{\tau^e,\lambda^e}[A_e = 1, X_e] \,:\, e \in \mathcal{E}\} \cup \{\mathbb{I}\left[X_{[d]} | \varnothing\right]\} \right\rangle_{\left[A_{\mathcal{E}}, X_{[d]}\right]} \\ &= \frac{\prod_{e\in\mathcal{E}} \lambda^e \left(\prod_{k\in e} m_k\right)}{\prod_{k\in[d]} m_k} \left\langle \{\tau^e[X_e] \,:\, e \in \mathcal{E}\} \right\rangle_{[X_{[d]}]} .\end{aligned}$$

Thus, $\tilde{\mathbb{P}}[A_{\mathcal{E}}, X_{[d]}]$ is an ancilla augmentation of the Markov Network with acceptance probability

$$\frac{\prod_{e\in\mathcal{E}} \lambda^e \left(\prod_{k\in e} m_k\right)}{\left\langle \{\tau^e[X_e] \,:\, e \in \mathcal{E}\} \right\rangle_{[\varnothing]} \cdot \left(\prod_{k\in[d]} m_k\right)} = \left(\prod_{k\in[d]} m_k\right)^{-1} \cdot \frac{\prod_{e\in\mathcal{E}} \left(\max_{x_e} \tau^e[X_e = x_e]\right)}{\left\langle \{\tau^e[X_e] \,:\, e \in \mathcal{E}\} \right\rangle_{[\varnothing]}} .$$

□

The main advantage of this augmentation is, that all cores are directed (see Figure 6b, and build therefore a Bayesian Network. Using the activation circuits studied above, we can now determine a rejection sampling procedure for arbitrary Markov Networks.

Lemma 5. *Let there be a Markov Network $\tau^{\mathcal{G}} = \{\tau^e[X_e] \,:\, e \in \mathcal{E}\}$ on a hypergraph $\mathcal{G}$. To each $e \in \mathcal{E}$ we build a function*

$$q_e = \frac{\tau^e[X_e = y_e]}{\max_{y_e} \tau^e[X_e = y_e]} .$$

We then build a concatenation of activation circuits to each function q_e, where each has a distinct ancilla qubit A_e. Then the constructed graph-controlled circuit acting on the state

$$\sqrt{\frac{1}{\prod_{v\in\mathcal{V}} m_v}} \mathbb{I}[X_{\mathcal{V}}] \otimes \left(\bigotimes_{e\in\mathcal{E}} \epsilon_0[A_e]\right)$$

prepares an ancilla augmented distribution of the Markov Network.

Proof. We show, that the measurement distribution is the ancilla augmented Markov Network. □

We use this result to prepare samples from Markov Networks by rejection sampling on the ancilla variables, in qcreason (see Sect. 6).

When the maximum of the hypercores are not known, and only an upper bound is given, it suffice to rescale them such that the maximum is lower than 1.

When the maximal coordinate of the Markov Network is, restricted to each edge, the maximal coordinate of the hypercore of that edge, we say that the Markov Network is not frustrated, otherwise frustrated.

Lemma 6. *If the Markov Network is not frustrated, then the acceptance probability λ of the rejection sampling scheme using the ancilla augmented Markov Network (with respect to the uniform distribution) is the ∞-Renyi divergence between the Markov Network and the uniform distribution.*

Proof. If and only if the Markov Network is not frustrated we have

$$\max_{x_{[d]}} \left(\prod_{e\in\mathcal{E}} \tau^e[X_e = x_e]\right) = \prod_{e\in\mathcal{E}} \left(\max_{x_e} \tau^e[X_e = x_e]\right) .$$

In this case, the ancilla augmentation has the best acceptance probability, since for the state $x_{[d]}$ maximizing all local cores we have

$$\tilde{\mathbb{P}}[A_{[p]} = 1_{[p]} | X_{[d]} = x_{[d]}] = 1 .$$

Alternatively: Show this with the acceptance rate in proof of Lem. 4, comparing with best naive augmentation. □

If the Markov Network is frustrated, we cannot achieve the best acceptance probability by the distributed ancilla preparation approach described above. In this case, one faces a tradeoff between the size of the circuit to prepare ancilla augmentations and the acceptance probability.

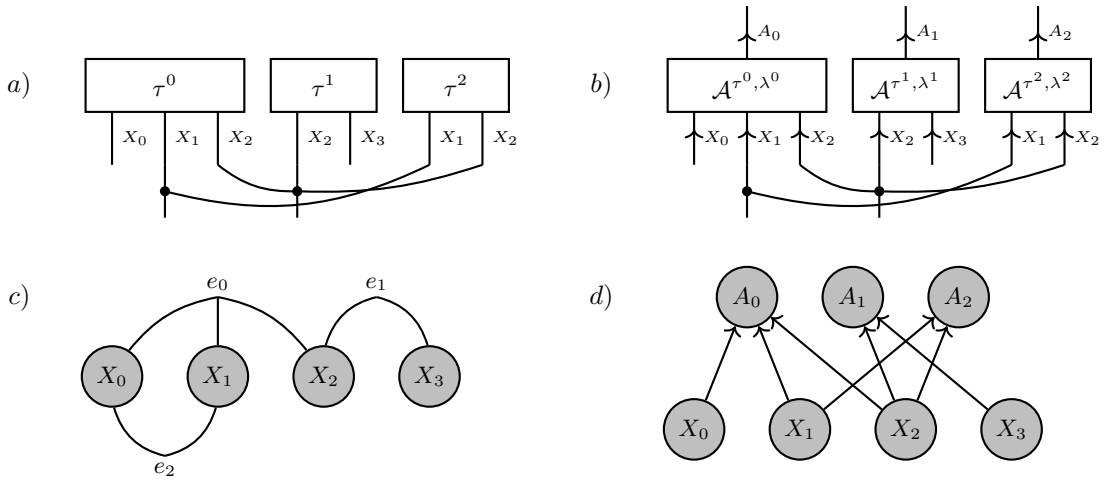


Figure 6: Ancilla augmentation of a Markov Network a) onto a Bayesian Network b). To this end the arbitrary non-negative cores τ^e are replaced by directed ancilla augmented cores $\tilde{\tau}^e$, with ancilly variables outgoing.

4 Computation circuits

We here suggest a quantum pendant to basis encodings (see Chapter 14 in Goessmann (2025)), which has the decomposition by contraction property.

Definition 5 (Computation circuit)**.** *Given a boolean function $f : \bigtimes_{k\in[d]}[2] \to [2]$ the computation circuit is the unitary tensor*

$$\begin{aligned}\mathcal{C}^f\left[Y_f^{\text{in}}, Y_f^{\text{out}}, X_{[d]}\right] \quad = & \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]\,:\,f\left[x_{[d]}\right]=1} \sigma_1\left[Y_f^{\text{in}}, Y_f^{\text{out}}\right] \otimes \epsilon_{x_{[d]}}\left[X_{[d]}\right] \\ & + \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]\,:\,f\left[x_{[d]}\right]=0} \delta\left[Y_f^{\text{in}}, Y_f^{\text{out}}\right] \otimes \epsilon_{x_{[d]}}\left[X_{[d]}\right] .\end{aligned}$$

Notice, that $\mathcal{C}^{\neg} = \text{CNOT}$, which is obvious from $\mathbb{I}\left[Y_{\text{in}}, Y_{\text{out}}\right] - \delta\left[Y_{\text{in}}, Y_{\text{out}}\right]$ being the Pauli-X gate (not to be confused with X denoting distributed variables here). The computation circuit is therefore a generalized controlled NOT gate, where the control is by a boolean function.

Functions with multiple output variables, i.e. $f : \bigtimes_{k\in[d]}[2] \to \bigtimes_{\ell\in[p]}[2]$, can be encoded image coordinate wise as a concatenation of the respective circuits.

4.1 Relation with basis encodings

Basis encodings are schemes to encode functions into directed boolean tensors, which enable reasoning about functions by tensor network contractions (see Goessmann et al. (2026)). To connect with this formalism, let us now show the relation to the computation circuits.

Remark 1 (Basis and Amplitude Encodings in Quantum Computation Schuld and Petruccione (2021) and tnreason)**.** *Basis Encoding in Quantum Computation refers to the representation of classical n bit strings by n qubit basis states, and is called one-hot encoding in* tnreason*. The Basis Encoding scheme in* tnreason *goes beyond this scheme and also encodes subsets by sums of one-hot encodings to the members of the set. In this way, relations and functions are represented by boolean tensors and contraction of them is refered as Basis calculus.*

Amplitude Encoding in Quantum Computation refers to the storage of complex numbers in the amplitudes of quantum states. The pendant in tnreason *is the Coordinate Encoding scheme, where real numbers are stored in the coordinates of real-valued tensors. Compared to Amplitude Encoding, Coordinate Encoding does not have the normalization constraint of quantum states. The Amplitude Encoding of the square root of a probability distribution is sometimes called q-sample.*

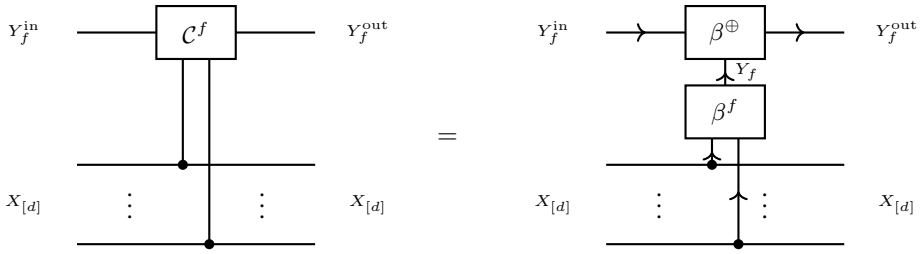


Figure 7: Relation of computation circuit and basis encodings. The computation circuit of a function is the contraction of its basis encoding with the basis encoding of the mod2 sum $\oplus$.

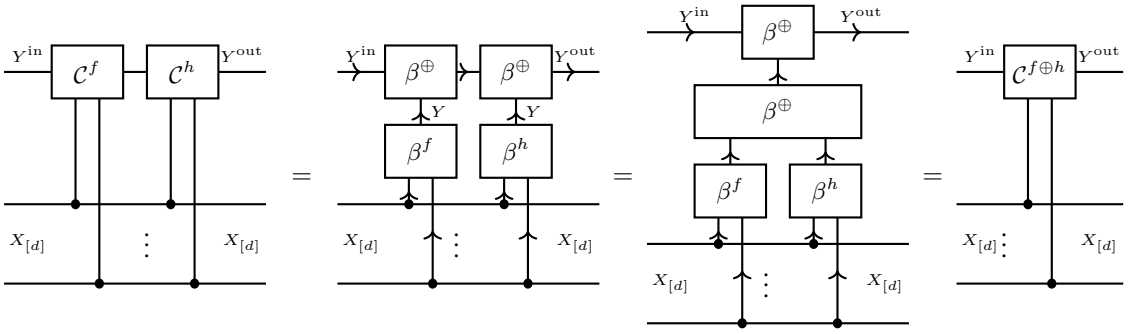


Figure 8: Equivalence of the composition of computation circuits with the computation circuit of their mod2 sum (Lem. 8).

Lemma 7. *For any boolean function we have (see Figure 7)*

$$\mathcal{C}^f\left[Y_f^{\text{in}}, Y_f^{\text{out}}, X_{[d]}\right] = \left\langle \beta^f\left[Y_f, X_{[d]}\right], \beta^{\oplus}\left[Y_f^{\text{out}}, Y_f^{\text{in}}, Y_f\right]\right\rangle_{\left[Y_f^{\text{in}}, Y_f^{\text{out}}, X_{[d]}\right]} .$$

In Goessmann et al. (2026) (and more detailed in Goessmann (2025)) sparse decompositions of basis encodings have been studied. Based on the equivalence of computation circuits and contracted basis encodings, stated by Lem. 7 we transfer these decomposition schemes in the following.

4.2 Construction for mod2-basis+ CP decompositions - Exploiting Polynomial sparsity

Lemma 8. *The composition of two computation circuits to functions q and g (see Figure 8), where the computation circuits share the target qubit, is the computation circuit of their sum mod2. In particular, computation circuits commute when they have the same target qubit.*

Proof. This follows from the representation of computation circuits by basis encodings of the corresponding function and the mod2 sum $\oplus$. □

Concatenating two computation circuits, which have the same head qubit, is the computation circuit of their mod2 sum.

Note that in general controlled unitaries on the same target qubit do not commute.

A basis+ elementary function can be encoded by a single controlled NOT operation with auxiliary X qubits.

This motivates the mod2-basis+ CP decomposition of tensors, which is exactly the decomposition of boolean polynomials into monomials. Each monomial is called a terms (products of x or (1-x) factors), and minterms in case that all variables appear.

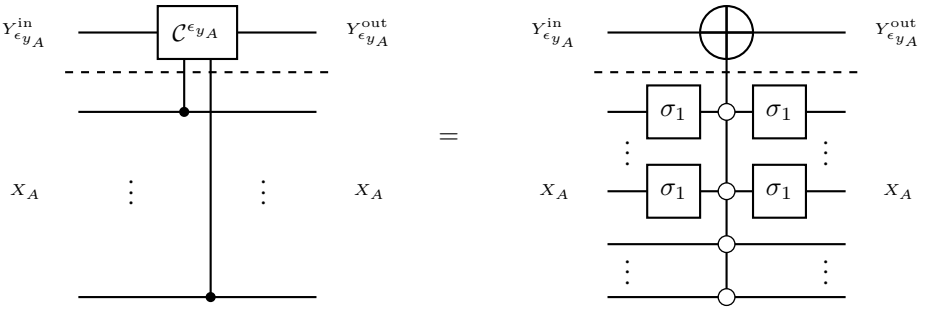


Figure 9: Exploitation of Polynomial sparsity in computation circuits.

Definition 6. *Given a boolean tensor τ, a mod2-basis+ CP decomposition is a collection $\mathcal{M}$ of tuples (A, x_A) with such that for any $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$*

$$\tau\left[X_{[d]} = x_{[d]}\right] = \bigoplus_{(A,x_A)\in\mathcal{M}} \langle \epsilon_{x_A}\left[X_A\right]\rangle_{\left[X_{[d]}=x_{[d]}\right]} .$$

Using that basis CP decompositions are a special case of basis+ CP decompositions, we get the following rank bound.

Lemma 9. *The mod2-basis+ CP rank is bounded by the basis CP rank.*

Proof. Use $A = [d]$, and x_A to each supported state. Then the mod2-sum is a usual sum and the basis CP decomposition is also a mod2-basis+ CP decomposition. □

This shows in particular, that any propositional formula can be represented by a mod2-basis+ CP decomposition.

Lemma 10. *The computation circuit to a boolean tensor τ with a mod2-basis+ CP decomposition $\mathcal{M}$ obeys*

$$\begin{aligned}&\mathcal{C}^{\tau}\left[Y^{\text{in}}_{\tau}, Y^{\text{out}}_{\tau}, X_{[d]}\right]\\ &\quad= \left\langle \{\delta\left[Y^{\text{in}}_{\tau}, Y_0\right], \delta\left[Y^{\text{out}}_{\tau}, Y_{|\mathcal{M}|-1}\right]\} \cup \{\mathcal{C}^{\epsilon_{x_A}}\left[Y_i, Y_{i+1}, X_A\right] : (A, x_A) \in \mathcal{M}\}\right\rangle_{\left[Y^{\text{in}}_{\tau}, Y^{\text{out}}_{\tau}, X_{[d]}\right]}\end{aligned}$$

where $i \in [|\mathcal{M}|]$ enumerates the tuples in $\mathcal{M}$.

Proof. Can be seen by the basis encoding representation, and the accociativtiy of the sum mod2 ($\oplus$). □

Each computation circuit to each boolean monomial can be prepared by a multiple-controlled σ_1 gate and further pairs of σ_1 gates preparing the control state, see Figure 9. When we sum monomials wrt modulus 2 calculus, then the preparation is a sequence of such circuits. In such way, we can prepare the computation circuit to any propositional formula. This encoding strategy exploits a modified (by mod2 calculus) polynomial sparsity.

4.3 Composition by Contraction - Exploiting Decomposition sparsity

The decomposition by contraction property of basis encodings is now a composition of circuits property, as stated in the next lemma. Instead of hidden variables as in the tnreason approach, we here exploit further working qubits.

Lemma 11. *We have for functions $q : \bigtimes_{k\in[d]}[2] \to \bigtimes_{\ell\in[p]}[2]$, $g : \bigtimes_{\ell\in[p]}[2] \to \bigtimes_{s\in[r]}[2]$ (see Figure 10)*

$$\begin{aligned}&\mathcal{C}^{g\circ q}\left[Y^{\text{in}}_{g\circ q}, Y^{\text{out}}_{g\circ q}, X_{[d]}\right] \otimes \epsilon_0\left[Y^{\text{out}}_{q}\right]\\ &\quad= \left\langle \epsilon_0\left[Y^{\text{in}}_{q}\right], \mathcal{C}^{q}\left[Y^{\text{in}}_{q}, Y^{\text{mid}}_{q}, X_{[d]}\right], \mathcal{C}^{g}\left[Y_{\text{in},g\circ q}, Y^{\text{out}}_{g\circ q}, Y^{\text{mid}}_{q}\right], \mathcal{C}^{q}\left[Y^{\text{mid}}_{q}, Y^{\text{out}}_{q}, X_{[d]}\right]\right\rangle_{\left[Y^{\text{in}}_{g\circ q}, Y^{\text{out}}_{g\circ q}, X_{[d]}, Y^{\text{out}}_{q}\right]} .\end{aligned}$$

Here each copy of $Y_{g\circ q}$ denotes a tuple of r boolean variables and Y_q of p boolean variables.

Proof. We show the claim based on basis calculus and sketch the steps in Figure 11. In the first equation we use Lem. 7 and represent each computation circuit by a contraction of two basis encodings. In the second equation we use the identity

$$\left\langle \beta^{\oplus}\left[Y^{\text{out}}, Y^{\text{in}}, Y_q\right], \epsilon_0\left[Y^{\text{in}}\right]\right\rangle_{[Y^{\text{out}}, Y_q]} = \delta\left[Y^{\text{out}}, Y_q\right] .$$

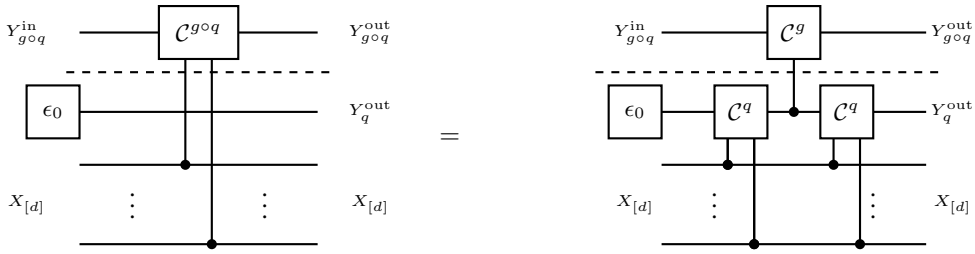


Figure 10: Exploitation of Decomposition sparsity in computation circuits. The working qubit Y_q is initialized in the ground state ϵ_0 and un-computed after its usage to prepare $Y_{g\circ q}$. It is then disentangled with the other qubits.

Now in the third equation we use that the contraction of basis encodings is the basis encoding of the composition function, and that $q \oplus q$ is the vanishing function, which basis encoding is a tensor product with $\epsilon_0 \left[Y_q^{\text{out}}\right]$. We use we Lem. 7 again in the fourth equation to arrive at the claim. □

When having a syntactical decomposition of a propositional formula, we can iteratively apply the computation circuit decomposition theorem and prepare each connective by a circuit. We can decompose any propositional formula into logical connectives and prepare to each a computation circuit (exploiting further sparsity mechanisms). This works, when the target qubit of one connective is used as a value qubit of another.

Note that to exploit decomposition sparsity, we introduce another auxiliary working qubit representing the satisfaction of the sub-function q. In Lem. 11 the equality of the concatenated computation circuits holds after contraction of this qubit, see also Figure 10. When performing a computational basis measurement on the state prepared by a concatenation, and omitting the qubit (either not measure it in the first place, or ignore its measurement), one effectively has closed the variable before measurement. This holds true due to the deterministic dependence of the measurement outcome on the measurement of the distributed qubits $X_{[d]}$, but is not true for more generic concatenations of controlled unitaries. This entailment of the working qubit with the distributed qubits can be resolved by an un-computation circuit, which is the adjoint of the computation circuit for the working qubits (see Figure 10). Note that while this is necessary in some applications, it is not necessary for the quantum rejection sampling presented here.

4.4 Preparation by fine and coarse structure

Having a mod2-basis+ CP decomposition of rank r to a connective, we need r controlled NOT gates to prepare the basis encoding. Given a syntactical decomposition of a boolean statistics, we prepare the basis encoding as a circuit with:

- **Fine Structure:** Represent each logical connective based on its mod2-basis+ CP decomposition, as a concatenation of computation circuits with the same variables.
- **Coarse Structure:** Arrange the logical connective representing circuits according to the syntactical hypergraph, where parent head variables appear as distributed variables at their children.

4.5 Preparation of function encoding states

Using the computation circuit we can prepare two function encoding schemes, depending on the start state.

4.5.1 Basis encoding

Definition 7. *The basis encoding of a boolean function f is the state*

$$\psi_f^{\text{bas}}\left[X_{[d]}, Y\right] = \sqrt{\frac{1}{2^d}} \sum_{x_{[d]} \in \times_{k\in[d]}[2]} \epsilon_{x_{[d]}}\left[X_{[d]}\right] \otimes \epsilon_{f(x_{[d]})}\left[Y\right] .$$

Lemma 12. *When applying the computation circuit on the initial state*

$$\sqrt{\frac{1}{2^d}} \mathbb{I}\left[X_{[d]}\right] \otimes \epsilon_0\left[Y\right]$$

we get the basis encoding state (see Figure 12a).

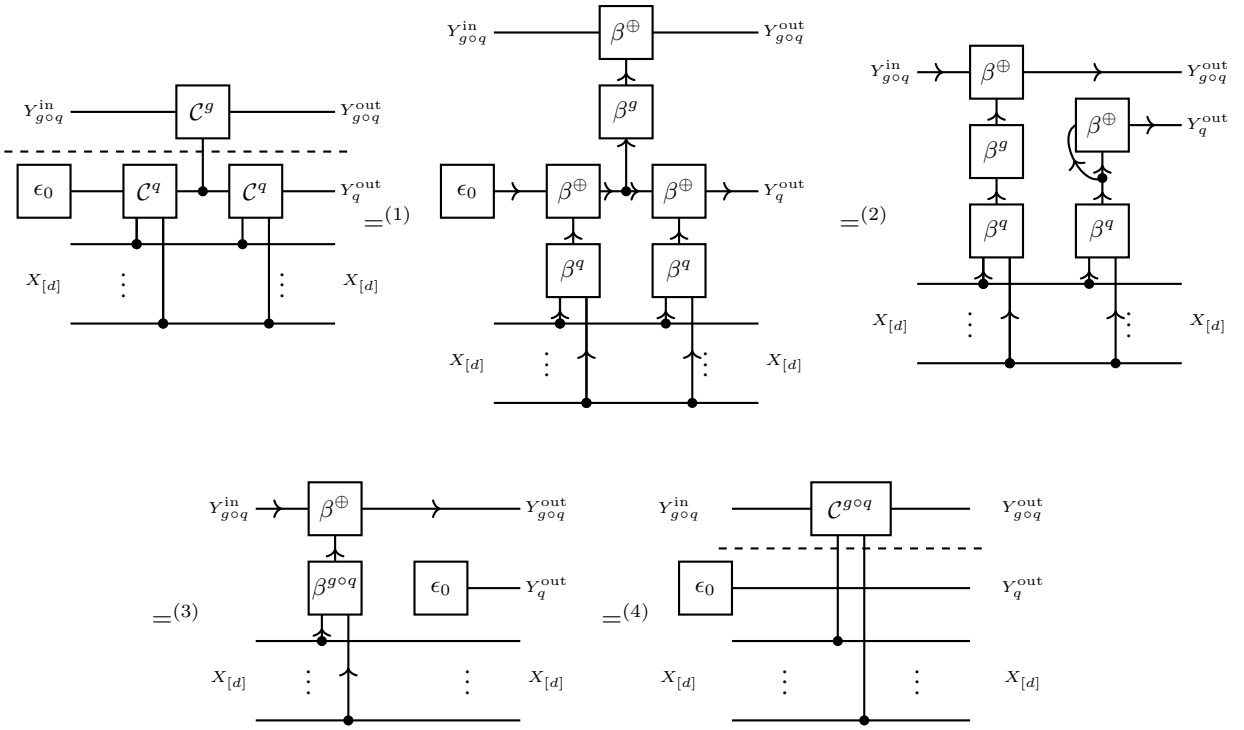


Figure 11: Proof of Lem. 11 based on basis calculus.

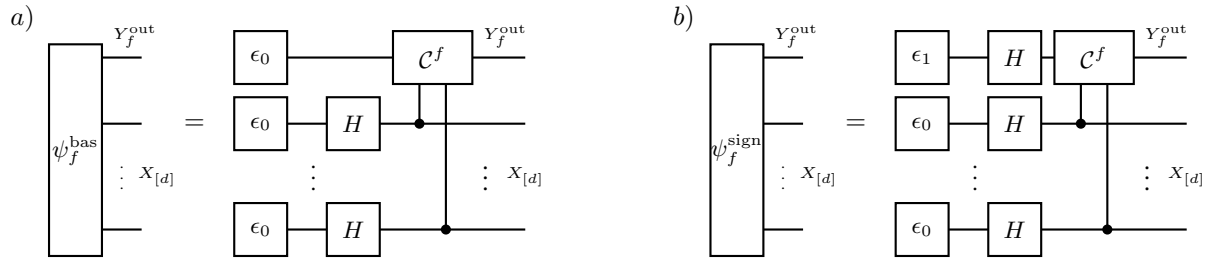


Figure 12: Construction of the basis encoding state (a) and the sign encoding state (b) using the computation circuit of a function.

Proof. Since the Pauli-X turns ϵ_0 into ϵ_1, and is applied exactly when $f(x_{[d]}) = 1$. □

Corollary 1. *Using Lem. 7 we have*

$$\psi^{\text{bas}}_f\left[X_{[d]}, Y\right] = \sqrt{\frac{1}{2^d}}\beta^f\left[X_{[d]}, Y\right]$$

The initial state can be prepared by applying Hadamard gates on all distributed variable qubits and preparing the head variable qubit in the ground state. The overlap of two basis encoding states (which share the same head variable) is proportional to the contraction of the corresponding formulas.

Lemma 13. *Let $f, \tilde{f}$ be two Boolean formulas on the states of $X_{[d]}$. Then we have*

$$\left\langle \psi^{\text{bas}}_f\left[X_{[d]}, Y\right], \psi^{\text{bas}}_h\left[X_{[d]}, Y\right]\right\rangle_{[\varnothing]} = \frac{1}{2^d}\left\langle f\left[X_{[d]}\right], \tilde{f}\left[X_{[d]}\right]\right\rangle_{[\varnothing]}.$$

Proof. We have

$$\begin{aligned}\left\langle \psi_f^{\mathrm{bas}}\left[X_{[d]},Y\right],\psi_h^{\mathrm{bas}}\left[X_{[d]},Y\right]\right\rangle_{[\varnothing]} &= \frac{1}{2^d}\sum_{x_{[d]}\in[2]^d}\left\langle \epsilon_{f\left[X_{[d]}=x_{[d]}\right]}\left[Y\right],\epsilon_{\tilde{f}\left[X_{[d]}=x_{[d]}\right]}\left[Y\right]\right\rangle_{[\varnothing]} \\ &= \frac{1}{2^d}\left\langle f\left[X_{[d]}\right],\tilde{f}\left[X_{[d]}\right]\right\rangle_{[\varnothing]} .\end{aligned}$$

□

4.5.2 Sign encoding

Definition 8. *The sign encoding of a boolean function f is the state*

$$\psi_f^{\mathrm{sign}}\left[X_{[d]},Y\right] = \left(\sqrt{\frac{1}{2^d}}\sum_{x_{[d]}\in\times_{k\in[d]}[2]}(-1)^{f(x_{[d]})}\cdot\epsilon_{x_{[d]}}\left[X_{[d]}\right]\right)\otimes\sqrt{\frac{1}{2}}\left(\epsilon_0\left[Y\right]-\epsilon_1\left[Y\right]\right) .$$

Sign encodings can be prepared by the phase kickback trick, as we show next.

Lemma 14 (Phase kickback trick)**.** *Applying the computation circuit $\mathcal{C}^f\left[Y^{\mathrm{in}},Y^{\mathrm{out}},X_{[d]}\right]$ on a state*

$$\psi\left[X_{[d]}\right]\otimes\epsilon^-\left[A^{\mathrm{in}}\right]$$

we get

$$\sum_{x_{[d]}\in\times_{k\in[d]}[m_k]}(-1)^{f(x_{[d]})}\psi\left[X_{[d]}=x_{[d]}\right]\cdot\epsilon_{x_{[d]}}\left[X_{[d]}\right]\otimes\epsilon^-\left[A^{\mathrm{out}}\right] .$$

Proof. We get a disentangled ancilla qubit, because $\epsilon^-\left[A^{\mathrm{in}}\right]$ is an eigenvector of the identity and the Pauli-X. For the identity, which is applied when $f(x_{[d]})=0$, the eigenvalue is 1. For the Pauli-X, which is applied when $f(x_{[d]})=0$, the eigenvalue is (-1). □

We get the sign encoding with the phase kickback trick, when we initialize with the q-sample of the uniform distribution.

Corollary 2. *When applying the computation circuit on the initial state*

$$\sqrt{\frac{1}{2^d}}\mathbb{I}\left[X_{[d]}\right]\otimes\sqrt{\frac{1}{2}}\left(\epsilon_0\left[Y\right]-\epsilon_1\left[Y\right]\right)$$

we get the sign encoding state (see Figure 12b). Note that the initial state is a Hadamard gate acting on the state $\epsilon_1\left[Y\right]$.

The sign encoding is also called the phase encoding. One can ask, whether we can use other ancilla states and get different phase kickbacks while keeping the ancilla disentangled. However, since we construct here computation circuit we are restricted to the eigenvectors of the Pauli-X gate, which is the Hadamard basis

$$\epsilon^-\left[Y\right]=\sqrt{\frac{1}{2}}\left(\epsilon_0\left[Y\right]-\epsilon_1\left[Y\right]\right) \quad , \quad \epsilon^+\left[Y\right] = \sqrt{\frac{1}{2}}\left(\epsilon_0\left[Y\right]+\epsilon_1\left[Y\right]\right)$$

with eigenvalues (-1) and 0.

The initial state can be prepared by applying Hadamard gates on all distributed variable qubits and preparing the head variable qubit by applying a Hadamard gate on the first basis state, i.e.

$$\frac{1}{\sqrt{2}}\left(\epsilon_1\left[Y_{\mathrm{out}}\right]-\epsilon_0\left[Y_{\mathrm{out}}\right]\right)=H\left[Y_{\mathrm{out}},Y_{\mathrm{in}}\right]\epsilon_0\left[Y_{\mathrm{in}}\right] ,$$

Lemma 15. *Let $f,\tilde{f}$ be two Boolean formulas on the states of $X_{[d]}$. Then we have*

$$\left\langle \psi_f^{\mathrm{sign}}\left[X_{[d]},Y\right],\psi_h^{\mathrm{sign}}\left[X_{[d]},Y\right]\right\rangle_{[\varnothing]} = \frac{1}{2^d}\left(\left\langle f\left[X_{[d]}\right],\tilde{f}\left[X_{[d]}\right]\right\rangle_{[\varnothing]}-\left\langle f\left[X_{[d]}\right],\neg\tilde{f}\left[X_{[d]}\right]\right\rangle_{[\varnothing]}\right) .$$

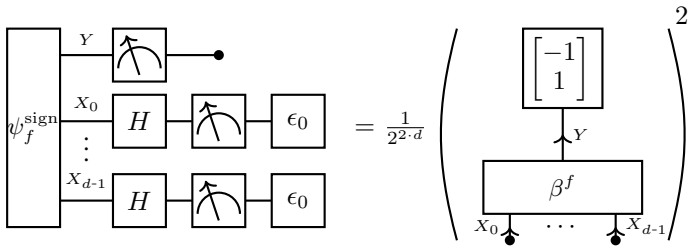


Figure 13: Measurement setup for the Deutsch-Jozsa algorithm on the formula f.

Proof. We have

$$\begin{aligned}\left\langle \psi_f^{\text{sign}}\left[X_{[d]},Y\right],\psi_h^{\text{sign}}\left[X_{[d]},Y\right]\right\rangle_{[\varnothing]} &= \frac{1}{2^d}\sum_{x_{[d]}\in[2]^d}(-1)^{f\left[X_{[d]}=x_{[d]}\right]+\tilde{f}\left[X_{[d]}=x_{[d]}\right]}\\ &= \frac{1}{2^d}\sum_{x_{[d]}\in[2]^d\,:\,f\left[X_{[d]}=x_{[d]}\right]=\tilde{f}\left[X_{[d]}=x_{[d]}\right]}1+\sum_{x_{[d]}\in[2]^d\,:\,f\left[X_{[d]}=x_{[d]}\right]\neq\tilde{f}\left[X_{[d]}=x_{[d]}\right]}(-1)\\ &= \frac{1}{2^d}\left(\left\langle f\left[X_{[d]}\right],\tilde{f}\left[X_{[d]}\right]\right\rangle_{[\varnothing]}-\left\langle f\left[X_{[d]}\right],\neg\tilde{f}\left[X_{[d]}\right]\right\rangle_{[\varnothing]}\right).\end{aligned}$$

□

Limitations:

- The prepared state ψ_f^{sign} is distinguished from the state $\psi_{\neg f}^{\text{sign}}$ only by a non-observable phase factor
$$\psi_f^{\text{sign}} = (-1)\cdot\psi_{\neg f}^{\text{sign}}$$
- With this construction the statistic qubit is disentangled with the distributed qubits. Further manipulation of the statistic qubits alone (as we do with activation circuits) will not retrieve any information about the encoded function.
- When doing the sign encoding procedure for multiple formulas, we get a disentangled state of the head variables with the distributed variables encoded in the sign encoding state of $\bigoplus_{f\in\mathcal{F}} f$ (where by $\bigoplus$ we denote the mod2 sum of boolean functions).

4.6 Applications

Let us present two known algorithms as the measurement of states prepared by computation circuits.

Example 6 (Deutsch-Jozsa Test)**.** *We measure the probability of the ground state of the sign encoding after a Walsh-Hadamard transform. The probability of measuring the ground state after a Walsh-Hadamard transform of the distributed qubits is then*

$$\frac{1}{2^d}\cdot\left\langle\left|\left\langle\psi_f^{\text{sign}}\left[X_{[d]},Y\right]\right\rangle_{[Y]}\right|^2\right\rangle_{[\varnothing]} = \frac{1}{2^{2\cdot d}}\left(\langle f\rangle_{[\varnothing]}-\langle\neg f\rangle_{[\varnothing]}\right)^2.$$

Thus, if and only if the probability of the ground state is 1*, we have that the formula is constant, that is $f\in\{\top,\bot\}$. This is the Deutsch-Josza algorithm Deutsch and Jozsa (1992).*

Example 7 (Inversion Test)**.** *Concatenating the computation circuits to two formulas f and h prepares the state*

$$\psi_{f\oplus h}^{\text{bas}}\left[X_{[d]},Y\right]$$

The probability of measuring the ground state after a Walsh-Hadamard transform of the distributed qubits (see Figure 14) is then

$$\begin{aligned}\frac{1}{2^d}\left|\left\langle\psi_{f\oplus h}^{\text{bas}}\left[X_{[d]},Y\right]\right\rangle_{[Y]}\right|^2 &= \frac{1}{2^{2\cdot d}}\left[\langle f\oplus h\rangle_{[\varnothing]}\left\langle\neg(f\oplus h)\right\rangle_{[\varnothing]}\right]\\ &= \frac{1}{2^{2\cdot d}}\begin{bmatrix}\#\left\{x_{[d]}\in\bigtimes_{k\in[d]}[2]\,:\,f\left[X_{[d]}=x_{[d]}\right]=h\left[X_{[d]}=x_{[d]}\right]\right\}\\ \#\left\{x_{[d]}\in\bigtimes_{k\in[d]}[2]\,:\,f\left[X_{[d]}=x_{[d]}\right]\neq h\left[X_{[d]}=x_{[d]}\right]\right\}\end{bmatrix}.\end{aligned}$$

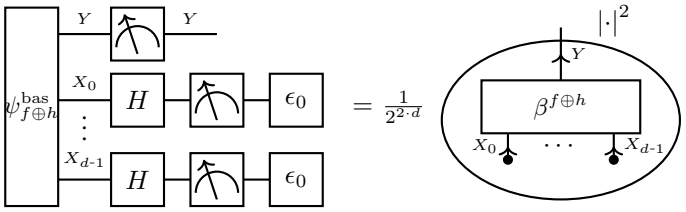


Figure 14: Measurement setup for the inversion test on the formulas f and h. We measure the probability of the statistic qubit Y, when the distributed qubits $X_{[d]}$ are in the ground state after a Walsh-Hadamard transform. This is equal to a scaled and squared contraction of the basis encoding to the formula $f \oplus h$.

The second equation holds since $\neg(f \oplus h)$ is equivalent with the biconditional $f \Leftrightarrow h$.

This is in more generality the inversion test of measuring the overlap of ψ^{bas}_f with ψ^{bas}_h (see Schuld and Petruccione (2021)). Here we used that the computation circuit is self-adjoint and understand the concatenation as the unitaries preparing ψ^{bas}_f and the adjoint (together with the Walsh-Hadamard transform) of ψ^{bas}_h.

Quantum Circuits such as the SWAP test and the Hadamard test ((Schuld and Petruccione, 2021, Section 3.6.1)) can be used to measure overlaps of quantum states, which are the squared absolutes of contractions of two state tensors.

- When we have preparation schemes for two tensors, we can control them with a common ancilla qubit and measure their contraction by a Hadamard test (alternatively, using the SWAP test and state preparation in two registers).
- Can we extend these schemes to contractions of more general tensor networks?

While the preparation of contracted basis encodings is done efficiently, accessing the prepared state by measurements is a bottleneck. This is a typical *quantum state tomography* problem Roth et al. (2023). Sparsity of the contracted state can be exploited to probabilistically guarantee precise estimates Gross et al. (2010). However, when deciding whether f is satisfiable, based on the prepared contraction $\sqrt{\langle \beta^f \left[Y, X_{[d]}\right] \rangle_{[Y|\varnothing]}}$, we have an exponentially small error tolerance. Moreover, the applied sample mean is the UMVUE (Uniformly Minimum Variance Unbiased Estimator, see Casella and Berger (2001)), and can thus not be improved by other unbiased estimators.

5 Sampling from Computation-Activation Networks

We investigate, how the above circuit encoding schemes can be applied in the preparation of states, which computational basis measurements are samples from specific distributions.

In particular, we build graph-controlled circuits being compositions of computation circuits and activation circuits, for which specific conditional distributions coincide with Computation-Activation Networks.

5.1 Preparing ancilla augmented distributions for Computation-Activation Networks

The computation network consists already of directed cores and therefore is already a Bayesian network. The activation network however needs ancilla augmentation. Therefore we have:

- Pendant for Coordinate Encoding in tnreason: Amplitude Encoding, storing the function value in the amplitude of an ancilla qubit. This is realized by an **Activation circuit** acting on an ancilla qubit in the ground state.
- Pendant for Basis Encoding in tnreason: **Computation circuit**, with composition by contraction property. Applied on the ground state, the computation circuit generates the basis encoding quantum state, which is parallel to the basis encoding.

Both are defined using controlled single qubit gates (see Sections 4.2-3 in Nielsen (2010)) with ancilla qubits being the target qubits.

Elementary Computation-Activation Networks can be augmented by ancilla augmentation of each leg vector in the activation tensor network (see Figure 15). We understand ancilla variables as variables in a Bayesian Network having single parents, corresponding with the ancilla augmentation of an elementary tensor. For an example, see Figure 18.

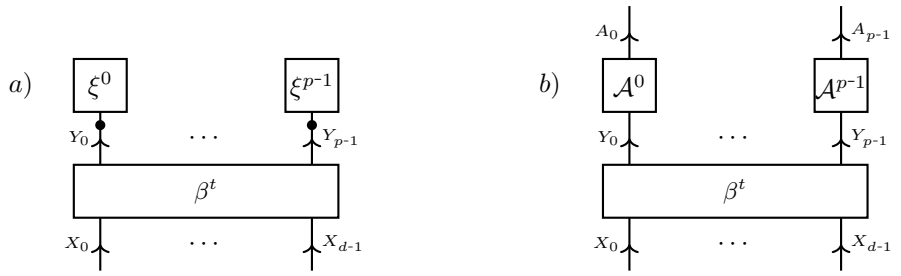


Figure 15: Ancilla augmentation of an elementary Computation-Activation Network a) by replacing each leg vector of the elementary activation tensor by an directed ancilla augmentation.

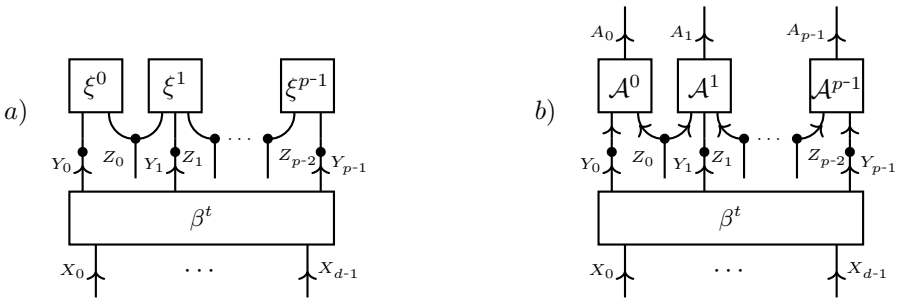


Figure 16: Ancilla augmentation of an TT Computation-Activation Network a) by a Bayesian Network b). The hidden variables in the TT activation tensor are treated as distributed variables. Marginalization over the hidden variables and conditioning on the activation variables in state 1, we reproduce the Computation-Activation Network.

To a generic activation hypergraph $\mathcal{G}$, we would do ancilla augmentation for the activation tensor network, where each hidden variable is treated as a distributed qubit. To treat it as a distributed qubit, each hidden activation qubit is prepared in uniform state (i.e. a Hadamard gate acting on a ground state) and will be omitted in the measurement scheme (either not measured, or its measurement ignored). This can be understood from a tensor network perspective, where marginalization means contracting, which is exactly how hidden activation variables are used. For an example in the TT format, see Figure 16.

5.2 Amplitude Amplification

The above scheme prepares the ancilla augmented Computation-Activation Network

$$\tilde{\mathbb{P}}[A_{[p]}, Y_{[p]}, X_{[d]}] = \lambda \cdot \epsilon_{1_{[p]}}\left[A_{[p]}\right] \otimes \left\langle \beta^t\left[Y_{[p]}, X_{[d]}\right], \mathbb{P}\left[X_{[d]}\right]\right\rangle_{\left[Y_{[p]}, X_{[d]}\right]} + (1-\lambda)\mathbb{P}^{\perp}\left[X_{[d]}, Y_{[p]}, A_{[p]}\right]$$

and the prepared state is the combination of q-samples

$$\psi^0\left[A_{[p]}, Y_{[p]}, X_{[d]}\right] = \sqrt{\lambda} \cdot \epsilon_{1_{[p]}}\left[A_{[p]}\right] \otimes \left\langle \beta^t\left[Y_{[p]}, X_{[d]}\right], \sqrt{\mathbb{P}\left[X_{[d]}\right]}\right\rangle_{\left[Y_{[p]}, X_{[d]}\right]} + \sqrt{(1-\lambda)}\sqrt{\mathbb{P}^{\perp}\left[X_{[d]}, Y_{[p]}, A_{[p]}\right]}\,.$$

In the subspace V spanned by the axes

- $\psi^t = \epsilon_{1_{[p]}}\left[A_{[p]}\right] \otimes \left\langle \beta^t\left[Y_{[p]}, X_{[d]}\right], \sqrt{\mathbb{P}\left[X_{[d]}\right]}\right\rangle_{\left[Y_{[p]}, X_{[d]}\right]}$
- $\psi^{\perp} = \sqrt{\mathbb{P}^{\perp}}$

which can be thought of the polar angle coordinate of the state

$$\psi^0\left[A_{[p]}, Y_{[p]}, X_{[d]}\right] = \sin\left(\frac{\alpha}{2}\right)\psi^0\left[A_{[p]}, Y_{[p]}, X_{[d]}\right] + \cos\left(\frac{\alpha}{2}\right)\psi^1\left[A_{[p]}, Y_{[p]}, X_{[d]}\right]$$

where we define an angle α by

$$\sin\left(\frac{\alpha}{2}\right) \coloneqq \sqrt{\lambda}$$

The amplitude amplification scheme now iterates between

- Reflection along ψ^t, by

$$\mathcal{S}^{\|}\left[A^{\text{in}}_{[p]}, A^{\text{out}}_{[p]}\right] = \delta\left[A^{\text{in}}_{[p]}, A^{\text{out}}_{[p]}\right] - 2\epsilon_{1_{[p]}}\left[A^{\text{in}}_{[p]}\right] \otimes \epsilon_{1_{[p]}}\left[A^{\text{out}}_{[p]}\right]$$

- Reflection along ψ, by

$$\begin{aligned}\mathcal{S}^{\psi}\left[A^{\text{in}}_{[p]}, Y^{\text{in}}_{[p]}, A^{\text{in}}_{[p]}, A^{\text{out}}_{[p]}, Y^{\text{out}}_{[p]}, A^{\text{out}}_{[p]}\right] =& \delta\left[A^{\text{in}}_{[p]}, Y^{\text{in}}_{[p]}, A^{\text{in}}_{[p]}, A^{\text{out}}_{[p]}, Y^{\text{out}}_{[p]}, A^{\text{out}}_{[p]}\right] \\ &- 2\psi^0\left[A^{\text{in}}_{[p]}, Y^{\text{in}}_{[p]}, X^{\text{in}}_{[d]}\right] \otimes \psi^0\left[A^{\text{out}}_{[p]}, Y^{\text{out}}_{[p]}, X^{\text{out}}_{[d]}\right]\end{aligned}$$

  This reflection can be done by concatenating the operator preparing the ancilla-augmentation of the Computation-Activation Network with a initial state reflection.

Now the application of j grover operators $S_0 S_\psi$ on ψ^0 prepares a state

$$\psi^j\left[A_{[p]}, Y_{[p]}, X_{[d]}\right] = \sin\left(\left(\frac{1}{2}+j\right)\alpha\right)\psi^{\|}\left[A_{[p]}, Y_{[p]}, X_{[d]}\right] + \sin\left(\left(\frac{1}{2}+j\right)\alpha\right)\psi^{\perp}\left[A_{[p]}, Y_{[p]}, X_{[d]}\right]$$

The state is closest to $\psi^{\|}$ when

$$\left(\frac{1}{2}+j\right)\alpha \approx \frac{\pi}{2}\,.$$

Estimating for small λ (tight for small λ)

$$\sqrt{\lambda} = \sin\left(\frac{\alpha}{2}\right) \leq \frac{\alpha}{2}$$

we get a bound

$$j^{\text{opt}} \leq \frac{\pi}{4\sqrt{\lambda}}\,.$$

This shows that we have a square root improvement on classical rejection sampling, which would require $\frac{1}{\lambda}$ repetitions to produce a sample. Note however that we compare quantum circuit length with the number of classical repetitions in sampling.

Literature:

- Grover (1996) Grover algorithm (search in unstructured database)
- Brassard and Hoyer (1997); Brassard et al. (2002) generalization to amplitude amplification
- Ozols et al. (2013) introduced quantum rejection sampling (using amplitude amplification)
- Low et al. (2014) used quantum rejection sampling for Bayesian network sampling (which is NP-hard when conditioned on evidence, see e.g. Koller and Friedman (2009))
- No exponential speedups expected by Quantum Computing in AI Bennett et al. (1997)

Note, that the variable qubits are uniformly distributed when only the computation circuit is applied. When sampling the probability distribution, we need the ancilla qubits to be in state 1 in order for the sample to be valid. Any other states will have to be rejected.

Classically, this can be simulated in the same way: Just draw the variables from uniform, calculate the value qubit by a logical circuit inference and accept with probability by the computed value.

For this procedure to be more effective (and in particular not having an efficient classical pendant), we need amplitude amplification on the value qubit. This can provide a square root speedup in the complexity compared with classical rejection sampling.

Open Question: Is there a way to avoid amplitude amplification and use a more direct circuit implementation of the activation network? - Cannot be the case, when the encoding is determined by the activation tensor alone: Needs to use the computated statistic as well. Negative result in (Nielsen, 2010, Section 6.6): When using multiple applications of computation circuit in combination with further unitaries, need at least as many applications as with amplitude amplification.

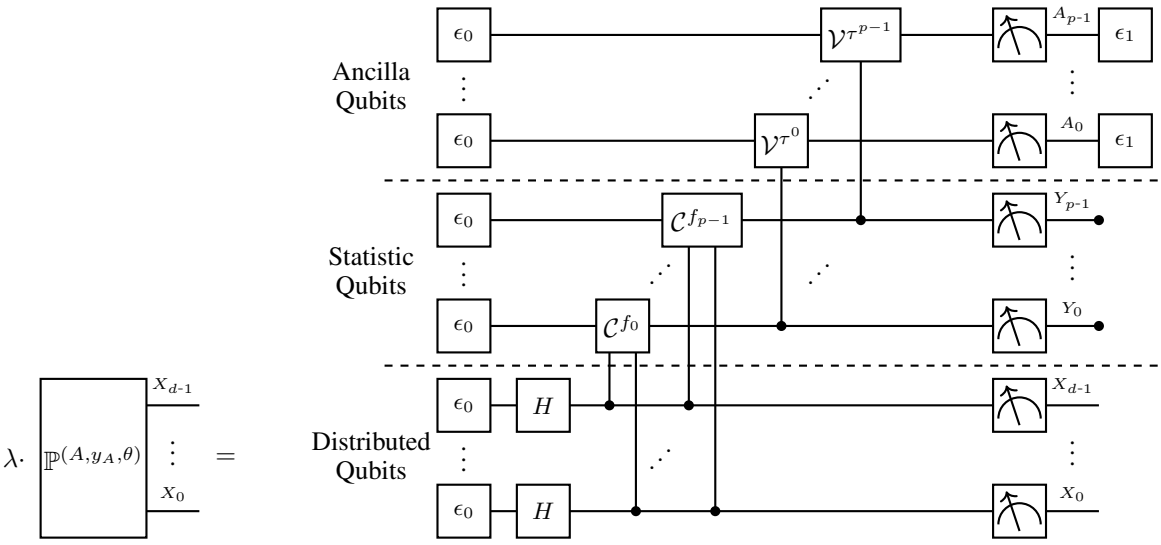


Figure 17: Quantum Circuit to reproduce a Computation-Activation Network (with elementary activation) by rejection sampling. We measure the distributed qubits $X_{[d]}$ and the ancilla qubits $A_{[p]}$ and reject all samples, where an ancilla qubit is measured as 0.

5.3 Sampling from Computation-Activation Networks as Quantum Circuits

So far: Sample from Hybrid Logic Networks, would need qudits for more general Computation-Activation Networks. Can do non-elementary Computation-Activation Networks, when activating whole activation core.

tnreason provides tensor network representations of knowledge bases and exponential families following a Computation Activation architecture. Here are some ideas to utilize quantum circuits for sampling from Computation-Activation Networks. We can produce Q-samples for ancilla augmented Computation-Activation Networks using computation circuits and activation circuits:

- For each (sub-) statistic, prepare a qubit by Computation circuits
- Based on the computed qubits, prepare ancilla qubits by Activation circuits to the activation cores.

Example 8 (Toy Accounting Example)**.** *For a more detailed example, let us consider a system of three variables* A1 *Account 1 is booked,* A2 *Account 2 is booked,* F *a feature on an invoice. Assume the following two rules have to be respected:*

- *Exactly one account must be booked.*
- *If feature* F *is present on the invoice, the account* A1 *is typically booked.*

We formalize this with the statistic

$$t = (X_{A1} \oplus X_{A2}, X_F \Rightarrow X_{A1})\,.$$

Any elementary Computation-Activation Network with the statistic t *can be realized by the circuit shown in Figure 18. For the preparation of the first statistic qubit* Y_0 *to* $t_0 = X_{A1} \oplus X_{A2}$ *by a computation circuit we exploit that*

$$\begin{aligned} X_{A1} \oplus X_{A2} &= \epsilon_1 [X_{A1}] \otimes \epsilon_0 [X_{A2}] + \epsilon_0 [X_{A1}] \otimes \epsilon_1 [X_{A2}] \\ &= (\epsilon_1 [X_{A1}] \otimes \epsilon_0 [X_{A2}]) \oplus (\epsilon_0 [X_{A1}] \otimes \epsilon_1 [X_{A2}]) \end{aligned}$$

where by $\oplus$ *we denote coordinatewise summation mod 2. The statistic qubit* Y_0 *is therefore prepared by two controlled NOT gates (see Figure 18b).*

For the preparation of the second statistic qubit Y_1 *to* $t_1 = X_F \Rightarrow X_{A1}$ *we exploit that the implication is* True *except for the case where the premise* X_F *is* True *and the head* X_{A1} *is* False*. In out mod 2 calculus, this amounts to*

$$X_F \Rightarrow X_{A1} = \mathbb{I}\,[X_F, X_{A1}] \oplus \epsilon_1 [X_F] \otimes \epsilon_0 [X_{A1}]\ .$$

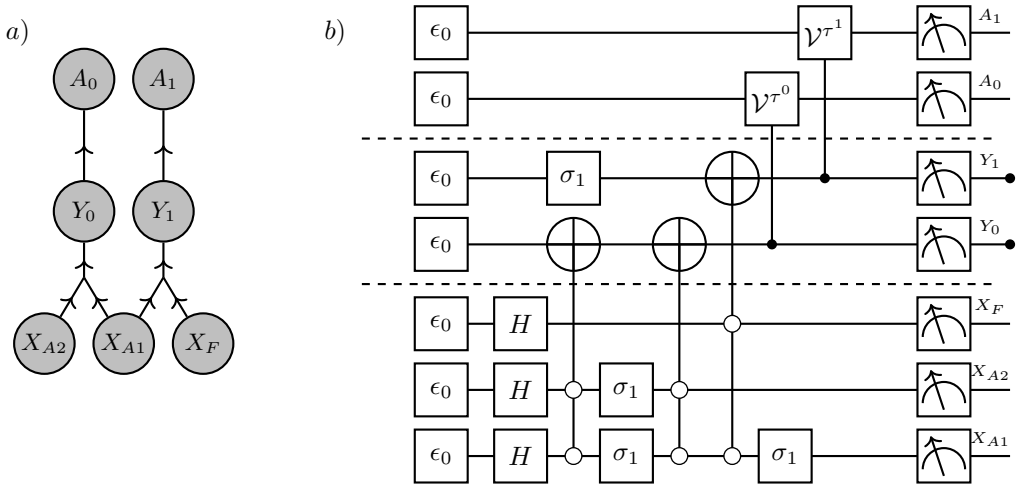


Figure 18: Circuit to sample from the Computation-Activation Network in the toy accounting Example 8.

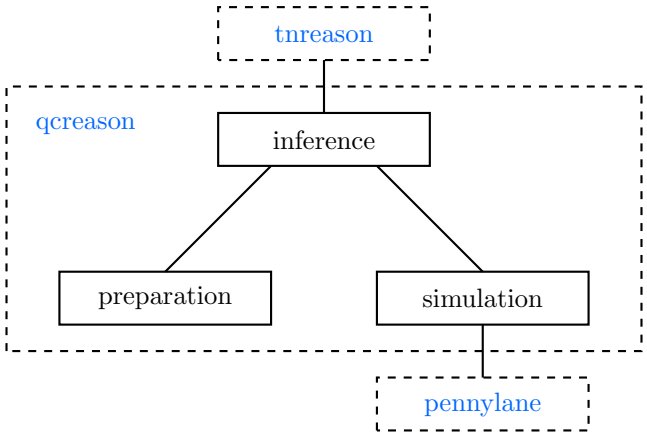


Figure 19: Architecture of the qcreason package. The inference sub-package receives inference tasks from tnreason and calls preparation and simulation for corresponding circuit preparation and simulation.

It follows that the statistic qubit Y_1 is prepared by a NOT gate (that is a Pauli-X gate σ_1) and a second controlled NOT gate (see Figure 18b).

For both cases, the preparation of the statistic qubit can be done by two controlled NOT gates exploiting the polynomial sparsity of the connectives.

Any non-elementary Computation-Activation Network with the statistic t can be prepared by the circuit, when choosing a single ancilla qubit, which is uniformly controlled by the qubits $Y_{[2]}$.

5.4 Application

When sampling from probability distributions, we can use these samples to estimate probabilistic queries. Building on such particle-based inference schemes, we can perform various inference schemes for Computation-Activation Networks, such as backward inference and message passing schemes.

6 Implementation

The introduced quantum circuit preparation schemes have been implemented in the python package qcreason, which architecture is sketched in Figure 19.

6.1 Quantum Circuits

Quantum Circuits are stored as lists of dictionaries, where each dictionary represents a quantum gate application. We specify the unitaries by strings accessible via the key *"unitary"*, as listed in the following table:

Unitary	String Representation	Parameters	Tensor Notation
Hadamard	*"H"*		$H\left[X^{\text{in}}, X^{\text{out}}\right]$
Pauli-X	*"X"*		$\sigma_1\left[A^{\text{in}}, A^{\text{out}}\right]$
Rotation around Y-axis	*"RY"*	Angle α	$R_Y(\alpha)\left[A^{\text{in}}, A^{\text{out}}\right]$

The control in controlled operations is specified by a dictionary accessible via the key *"control"*. It contains as keys the control qubits and their control states as values (i.e. the state in which the unitary is applied).

Further parameters such as angles are added in another dictionary, accessible via the key *"parameters"*.

For example a hadamard gate on qubit *"blue"* and a multiple controlled rotation around the Y-axis on qubit *"ancilla_c1"* controlled by qubits *"red"* and *"blue"* in state $(0, 1)$ with an angle is represented as:

```
[{'unitary': 'H',
    'target': ['blue']},
{'unitary': 'MCRY',
    'target': ['ancilla_c1'],
    'control': {'red': 1, 'blue': 0},
    'parameters': {'angle': np.float64(0.6121442808462428)}}]
```

Note that we omit the notation of incoming and outgoing variables.

6.2 Generic Contraction

So far, the generic contraction routine consists in activation circuits preparing for each tensor an ancilla variable, and post-selecting the measurements where the ancilla variable is 1. The resulting tensor of the contraction is a PandasCore storing the post-selected measurement results in a dataframe.

7 Conclusion and Outlook

We have characterized an intuitive class of Quantum Circuits, namely graph-controlled circuits. we have shown that their prepared distributions are exactly the Bayesian Networks on the corresponding hypergraph. Along this class the demonstrated rejection sampling schemes by Computation Activation Circuits achieve the best acceptance probability. Using amplitude amplification, we get a square root improvement of the quantum rejection sampling scheme compared with classical rejection sampling.

To improve on this quantum speedup, different circuit designs are necessary. We could allow for hypergraphs with cycles, where e.g. the distributed qubits are rotated conditioned on the statistic qubits. Such schemes shall be investigated in future research. However, an improvement beyond scaling constants is not expected, since the amplitude amplification procedure already achieves the Heisenberg limit Brassard et al. (2002); **?**.

We have seen that preparing a classically hard to compute contraction can be done efficiently by a quantum circuit. We showed in particular that contracted basis encodings can be prepared by circuits, which are classically hard to contract in the presence of undirected loops in the decomposition hypergraph. However, when assessing the prepared contraction state by computational basis measurements, we effectively draw samples from the corresponding distribution, which can also be achieved classically. A quantum advantage could arise from different measurement schemes studied in the quantum state tomography community (see Gross et al. (2010); Cramer et al. (2010); Roth et al. (2023)). To this end, future research could show, how hard to achieve expectation queries can efficiently be prepared by customized measurement schemes.

One might further investigate the relation between amplitude amplification and change of the canonical parameters in Hybrid Logic Network. This will enable a more integrated alternative to the Parameter Estimation routines suggested here. Furthermore it will enable the preparation of Computation-Activation Networks without the need of ancilla augmentation. While we expect limits due to the discrete degrees of freedom of amplitude amplification, we could refine prepared circuits based on ancilla augmentation. To this end, we would understand such preparation as a coarse representation of Computation-Activation Networks, which serve as initial states for ancilla augmentation. The

advantage then lies in the improvement of the λ, when the ∞ Renyi divergence with respect to the coarse distribution is less than the ∞ Renyi divergence with respect to the uniform distribution.

A Extension: Sampling from proposal distributions

We can prepare basis circuit encodings to selection augmented formulas, in this way introducing formula selecting networks.

Idea for an inductive reasoning scheme: Prepare a q-sample from the empirical distribution and the current distribution. Then prepare the basis circuit encodings, where the selection variables are shared and the distributed variables assigned to the prepared samples. Now, the ancilla qubits can be designed to ϵ_1 and ϵ_0 accordingly. The rejection sampling scheme on both ancillas being 1 and the measurement of L prepares then the distribution

$$\left\langle \left\langle \mathbb{P}^D \left[X_{[d]}\right], \sigma^t \left[X_{[d]}, L\right] \right\rangle_{[L]}, \left\langle \tilde{\mathbb{P}} \left[X_{[d]}\right] \right\rangle_{[L]} \right\rangle_{[L|\varnothing]}$$

That is, the probability of selecting ℓ is proportional to

$$\mu_D \left[L = \ell\right] \cdot \left(1 - \tilde{\mu} \left[L = \ell\right]\right)$$

and thus prefers formulas, which have a large empirical mean, but a small current mean.

Open Question: Since the distribution is "similar" to $\exp\left[\mu_D \left[L = \ell\right] - \tilde{\mu} \left[L = \ell\right]\right]$ (terms appear in Taylor of first order), can we tune the distribution with an inverse temperature parameter β?

B Walsh-Hadamard transform

The application of Hadamard gates on the distributed variables afterwards results in a Walsh-Hadamard transform of the prepared state.

B.1 Transformation tensor

The transform is performed by applying Hadamard gates on each variable. This is equivalent with contraction of the tensor

$$H^d \left[X_{[d],\mathrm{in}}, X_{[d],\mathrm{out}}\right] = \bigotimes_{k \in [d]} H \left[X_k^{\mathrm{in}}, X_k^{\mathrm{out}}\right] ,$$

which coordinates are

$$H^d \left[X_{[d],\mathrm{in}} = x_{[d],\mathrm{in}}, X_{[d],\mathrm{out}} = x_{[d],\mathrm{out}}\right] = \sqrt{\frac{1}{2^d}} \, (-1)^{\sum_{k \in [d]} x_{k,\mathrm{in}} \cdot x_{k,\mathrm{out}}} .$$

We have

$$\left\langle \epsilon_0 \left[X^{\mathrm{in}}\right], H \left[X^{\mathrm{in}}, X^{\mathrm{out}}\right] \right\rangle_{[X^{\mathrm{out}}]} = \sqrt{\frac{1}{2}} \mathbb{I} \left[X^{\mathrm{out}}\right]$$

$$\left\langle \epsilon_1 \left[X^{\mathrm{in}}\right], H \left[X^{\mathrm{in}}, X^{\mathrm{out}}\right] \right\rangle_{[X^{\mathrm{out}}]} = \sqrt{\frac{1}{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \left[X^{\mathrm{out}}\right]$$

and conversely

$$\left\langle \sqrt{\frac{1}{2}} \mathbb{I} \left[X^{\mathrm{in}}\right], H \left[X^{\mathrm{in}}, X^{\mathrm{out}}\right] \right\rangle_{[X^{\mathrm{out}}]} = \epsilon_0 \left[X^{\mathrm{out}}\right]$$

$$\left\langle \sqrt{\frac{1}{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \left[X^{\mathrm{in}}\right], H \left[X^{\mathrm{in}}, X^{\mathrm{out}}\right] \right\rangle_{[X^{\mathrm{out}}]} = \epsilon_1 \left[X^{\mathrm{out}}\right] .$$

If and only if the function is constant, then the transformed state is the ground state.

References

Israel F. Araujo, Carsten Blank, Ismael CS Araújo, and Adenilton J. da Silva. Low-rank quantum state preparation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(1):161–170, 2023. URL https://ieeexplore.ieee.org/abstract/document/10190145/.

Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing*, 26(5):1510–1523, October 1997. ISSN 0097-5397, 1095-7111. doi: 10.1137/S0097539796300933. URL http://arxiv.org/abs/quant-ph/9701001. arXiv:quant-ph/9701001.

Gilles Brassard and Peter Hoyer. An Exact Quantum Polynomial-Time Algorithm for Simon's Problem. In *Proceedings of the Fifth Israel Symposium on the Theory of Computing Systems (ISTCS '97)*, ISTCS '97, page 12, USA, June 1997. IEEE Computer Society. ISBN 978-0-8186-8037-3.

Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum Amplitude Amplification and Estimation, 2002. URL http://arxiv.org/abs/quant-ph/0005055. arXiv:quant-ph/0005055.

George Casella and Roger Berger. *Statistical Inference*. Cengage Learning, Pacific Grove, Calif, June 2001. ISBN 978-0-534-24312-8.

Marcus Cramer, Martin B. Plenio, Steven T. Flammia, Rolando Somma, David Gross, Stephen D. Bartlett, Olivier Landon-Cardinal, David Poulin, and Yi-Kai Liu. Efficient quantum state tomography. *Nature Communications*, 1 (1):149, December 2010. ISSN 2041-1723. doi: 10.1038/ncomms1147. URL https://www.nature.com/articles/ncomms1147.

David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, December 1992. ISSN 0962-8444. doi: 10.1098/rspa.1992.0167. URL https://doi.org/10.1098/rspa.1992.0167.

Alex Goessmann. The Tensor-Network Approach to Efficient and Explainable AI, 2025. URL https://github.com/EnexaProject/enexa-tensor-reasoning-documentation/.

Alex Goessmann, Janina Schütte, Maximilian Fröhlich, and Martin Eigel. A tensor network formalism for neuro-symbolic AI, January 2026. URL http://arxiv.org/abs/2601.15442. arXiv:2601.15442 [cs].

David Gross, Yi-Kai Liu, Steven T. Flammia, Stephen Becker, and Jens Eisert. Quantum State Tomography via Compressed Sensing. *Physical Review Letters*, 105(15):150401, October 2010. doi: 10.1103/PhysRevLett.105.150401. URL https://link.aps.org/doi/10.1103/PhysRevLett.105.150401.

Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions, August 2002. URL http://arxiv.org/abs/quant-ph/0208112. arXiv:quant-ph/0208112.

Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, July 1996. Association for Computing Machinery. ISBN 978-0-89791-785-8. doi: 10.1145/237814.237866. URL https://dl.acm.org/doi/10.1145/237814.237866.

Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum Algorithm for Linear Systems of Equations. *Physical Review Letters*, 103(15):150502, October 2009. doi: 10.1103/PhysRevLett.103.150502. URL https://link.aps.org/doi/10.1103/PhysRevLett.103.150502.

Steven Herbert. No quantum speedup with Grover-Rudolph state preparation for quantum Monte Carlo integration. *Physical Review E*, 103(6):063302, June 2021. doi: 10.1103/PhysRevE.103.063302. URL https://link.aps.org/doi/10.1103/PhysRevE.103.063302.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, Cambridge, Mass., 1. edition edition, July 2009. ISBN 978-0-262-01319-2.

Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang. Quantum inference on Bayesian networks. *Physical Review A*, 89(6):062315, June 2014. doi: 10.1103/PhysRevA.89.062315. URL https://link.aps.org/doi/10.1103/PhysRevA.89.062315.

Mikko Möttönen, Juha J. Vartiainen, Ville Bergholm, and Martti M. Salomaa. Transformation of quantum states using uniformly controlled rotations. *Quantum Info. Comput.*, 5(6):467–473, September 2005. ISSN 1533-7146.

Isaac L. Chuang Michael A. Nielsen. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, Cambridge ; New York, December 2010. ISBN 978-1-107-00217-3.

Maris Ozols, Martin Roetteler, and Jérémie Roland. Quantum rejection sampling. *ACM Trans. Comput. Theory*, 5(3): 11:1–11:33, August 2013. ISSN 1942-3454. doi: 10.1145/2493252.2493256. URL https://doi.org/10.1145/2493252.2493256.

John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018. doi: 10.22331/q-2018-08-06-79. URL https://quantum-journal.org/papers/q-2018-08-06-79/.

Ingo Roth, Jadwiga Wilkens, Dominik Hangleiter, and Jens Eisert. Semi-device-dependent blind quantum tomography. *Quantum*, 7:1053, July 2023. doi: 10.22331/q-2023-07-11-1053. URL https://quantum-journal.org/papers/q-2023-07-11-1053/.

Manuel S. Rudolph, Jing Chen, Jacob Miller, Atithi Acharya, and Alejandro Perdomo-Ortiz. Decomposition of matrix product states into shallow quantum circuits. *Quantum Science and Technology*, 9(1):015012, 2023a. URL https://iopscience.iop.org/article/10.1088/2058-9565/ad04e6/meta.

Manuel S. Rudolph, Jacob Miller, Danial Motlagh, Jing Chen, Atithi Acharya, and Alejandro Perdomo-Ortiz. Synergistic pretraining of parametrized quantum circuits via tensor networks. *Nature Communications*, 14(1):8367, 2023b. URL https://www.nature.com/articles/s41467-023-43908-6.

Aaron Sander, Maximilian Fröhlich, Mazen Ali, Martin Eigel, Jens Eisert, Michael Hintermüller, Christian B. Mendl, Richard M. Milbradt, and Robert Wille. Quantum circuit simulation with a local time-dependent variational principle, August 2025a. URL http://arxiv.org/abs/2508.10096. arXiv:2508.10096 [quant-ph].

Aaron Sander, Maximilian Fröhlich, Martin Eigel, Jens Eisert, Patrick Gelß, Michael Hintermüller, Richard M. Milbradt, Robert Wille, and Christian B. Mendl. Large-scale stochastic simulation of open quantum systems. *Nature Communications*, 16(1):11074, December 2025b. ISSN 2041-1723. doi: 10.1038/s41467-025-66846-x. URL http://arxiv.org/abs/2501.17913. arXiv:2501.17913 [quant-ph].

Maria Schuld and Francesco Petruccione. *Machine Learning with Quantum Computers*. Springer, Cham, October 2021. ISBN 978-3-030-83097-7.

Vivek V. Shende, Stephen S. Bullock, and Igor L. Markov. Synthesis of Quantum Logic Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010, June 2006. ISSN 0278-0070, 1937-4151. doi: 10.1109/TCAD.2005.855930. URL http://arxiv.org/abs/quant-ph/0406176. arXiv:quant-ph/0406176.

Peter Wittek and Christian Gogolin. Quantum Enhanced Inference in Markov Logic Networks. *Scientific Reports*, 7(1): 45672, April 2017. ISSN 2045-2322. doi: 10.1038/srep45672. URL http://arxiv.org/abs/1611.08104.