NEURO-SYMBOLIC QUANTUM INFERENCE

RESEARCH NOTES IN THE ENEXA AND QROM PROJECTS

January 20, 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

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.

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-Josza algorithm, which we generalize here.

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)

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.

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

1.1 Related Works

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)

Quantum algorithms for linear algebra:

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

2 Quantum Computation Basics

2.1 State Encoding Schemes

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.

2.2 Controlled Single Qubit Gates

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

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

Further we define the Pauli-X:

$$\sigma_1[A_{\text{in}}, A_{\text{out}}] \coloneqq \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

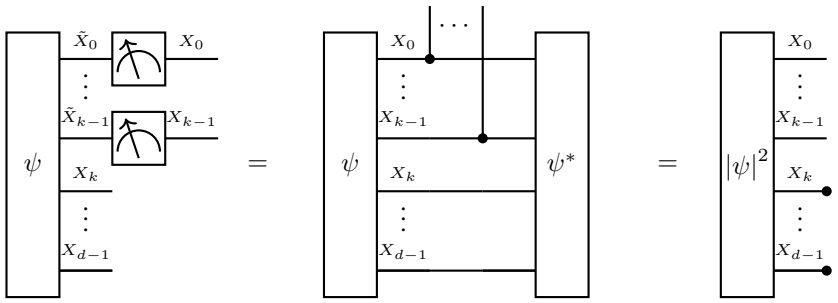


Figure 1: Computational Basis Measurement of a quantum state ψ. The measurement symbols on the left side indicate the measured qubits and the first equation is understood as a definition. In the second equation we sketch, that the measurement distribution is equal to the contraction of the square absolute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

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

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

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

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

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

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

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

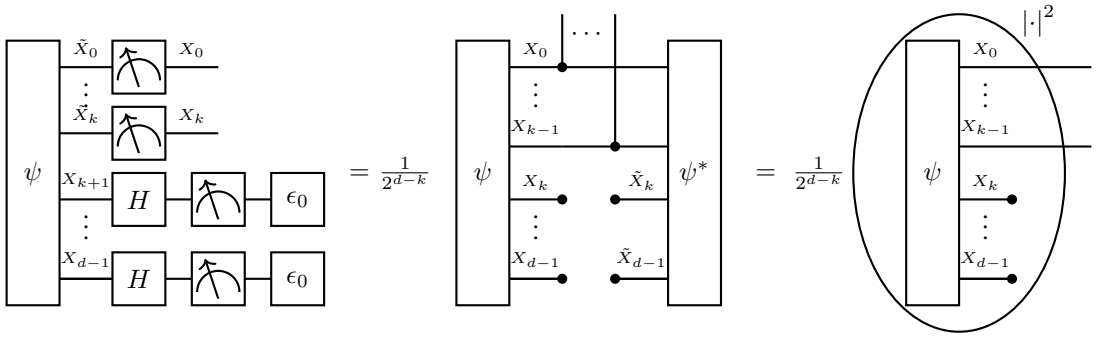


Figure 2: Computational Basis Measurement of a quantum state ψ, after the Walsh-Hadamard Transform on the closed qubits. By the ellipse we indicate a coordinatewise transform to the square of the absolute, after the contraction.

2.3.2 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.

2.4 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.

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*

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

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

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 1. *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 $\text{Pa}(k) \subset [k]$. 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 in Sect. 3. 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 $\mathbb{P}\left[X_{[d]}\right]$. □

The contrary is true, also when nodes appear multiple times as outgoing nodes.

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*

$$\big(X_v \perp X_{\mathrm{NonDes}(v)}\big) \;\big|\; X_{\mathrm{Pa}(v)}\,.$$

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

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

which establishes the conditional independence $\big(X_v \perp X_{\mathrm{NonDes}(v)}\big) \;\big|\; X_{\mathrm{Pa}(v)}$. □

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

Example 2 (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 Function Encoding Schemes

We investigate here quantum pendants to the function encoding schemes used in tnreason. All the schemes are graph-controlled circuits.

- 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, Chuang]) with ancilla qubits being the target qubits.

3.1 Activation circuit

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 $h(\cdot)$.

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

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

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

$$f : \bigtimes_{\ell\in[p]} [2] \to [0,1]$$

its activation circuit is the uniformly controlled unitary $\mathcal{V}^\tau\left[A_{\mathrm{in}}, A_{\mathrm{out}}, Y_{[p]}\right]$ *defined as*

$$\mathcal{V}^\tau\left[A_{\mathrm{in}}, A_{\mathrm{out}}, Y_{[p]}\right] = \sum_{y_{[p]}} \epsilon_{y_{[p]}}\left[Y_{[p]}\right] \otimes R_Y\left(h(f\left[y_{[p]}\right])\right)\left[A_{\mathrm{in}}, A_{\mathrm{out}}\right] .$$

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 $\mathcal{V} = Y_{[p]} \cup \{A\}$ and a single edge $\mathcal{E} = (Y_{[p]}, \{A\})$.

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.

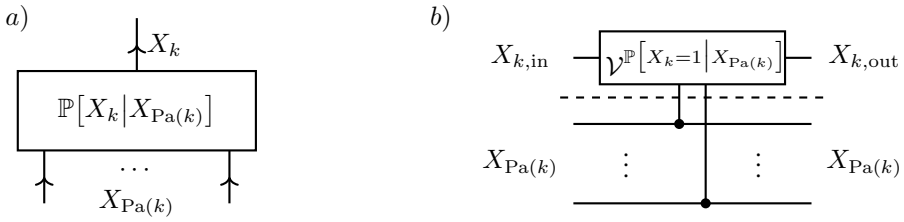


Figure 3: Representation of directed and positive tensor by a controlled rotation. a) Conditional probability tensor $\mathbb{P}\left[X_k \middle| X_{\mathrm{Pa}(k)}\right]$ being a tensor in a Bayesian Network. b) Circuit Encoding as a controlled rotation, which is the Activation circuit of the tensor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

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

$$f\left[y_{[p]}\right] = \frac{\tau\left[Y_{[p]} = y_{[p]}\right]}{\max_{\tilde{y}_{[p]}} \tau\left[Y_{[p]} = \tilde{y}_{[p]}\right]}$$

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).

3.1.1 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)

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

3.1.2 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 2 (Low et al.)**.** *Any Bayesian Network of variables $X_{[d]}$, where the enumeration by $[d]$ respects the partial order by child-parent relations, can be prepared as a quantum circuit by concatenating the activation circuits*

$$\mathcal{V}^{\mathbb{P}\left[X_k=1 \middle| X_{\mathrm{Pa}(k)}\right]}\left[X_{k,\mathrm{in}}, X_k, X_{\mathrm{Pa}(k)}, X_{\mathrm{Pa}(k)}\right]$$

for $k \in [d]$ and acting on the initial state $\bigotimes_{k\in[d]} \epsilon_0\left[X_{k,\mathrm{in}}\right]$.

3.2 Computation circuits

We here suggest a quantum pendant to basis encodings (see Chapter Basis Calculus), which has the decomposition by contraction property.

Definition 3 (Computation circuit)**.** *Given a boolean function $f : \bigtimes_{k\in[d]}[2] \to [2]$ the computation circuit is the unitary tensor*

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

Notice, that $\mathcal{C}^{\neg} = \mathrm{CNOT}$, which is obvious from $\mathbb{I}\left[Y_{\mathrm{in}}, Y_{\mathrm{out}}\right] - \delta\left[Y_{\mathrm{in}}, Y_{\mathrm{out}}\right]$ 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.

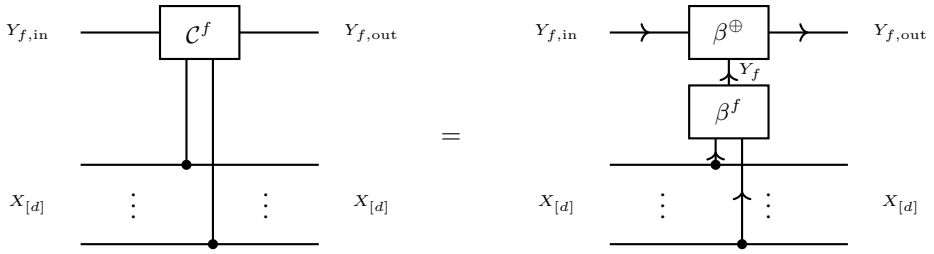


Figure 4: Equivalence of computation circuit and basis encodings.

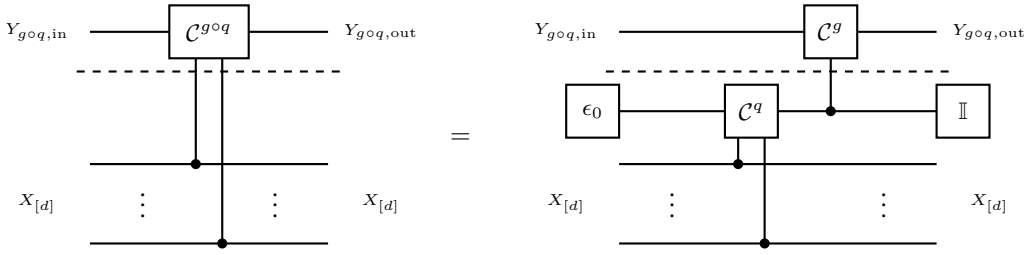


Figure 5: Exploitaition of Decomposition sparsity in computation circuits.

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.

3.2.1 Relation with basis encodings

Lemma 3. *For any boolean function we have (see Figure 4)*

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

3.2.2 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.

Lemma 4. *We have for functions $f : \bigtimes_{k\in[d]}[2] \to \bigtimes_{\ell\in[p]}[2]$, $g : \bigtimes_{\ell\in[p]}[2] \to \bigtimes_{s\in[r]}[2]$ (see Figure 5)*

$$\begin{aligned}\mathcal{C}^{g\circ f}\left[Y_{\text{in},g\circ f}, Y_{\text{out},g\circ f}, X_{\text{in},[d]}, X_{\text{out},[d]}\right] = \big\langle &\epsilon_0\left[Y_{\text{in},f}\right], \\ &\mathcal{C}^f\left[Y_{\text{in},f}, Y_{\text{out},f}, X_{\text{in},[d]}, X_{\text{out},[d]}\right], \\ &\mathcal{C}^g\left[Y_{\text{in},g\circ f}, Y_{\text{out},g\circ f}, Y_{\text{out},f}, Y_{\text{out},f}\right]\big\rangle_{\left[Y_{\text{in},g\circ f}, Y_{\text{out},g\circ f}, X_{\text{in},[d]}, X_{\text{out},[d]}\right]} .\end{aligned}$$

Proof. Can be seen by the basis encoding representation, when starting the hidden qubit with ϵ_0, since

$$\left\langle \beta^{\oplus}\left[Y, X_0, X_1\right], \epsilon_0\left[X_0\right]\right\rangle_{[Y,X_1]} = \delta\left[Y, X_1\right] .$$

The marginalization of the hidden target qubit X_1 for measurements does not change the distribution, since we have a deterministic dependence on the measured data register (need preparation by ϵ_0 for that). □

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 modulus 2 circuit implementation. This works, when the target qubit of one connective is used as a value qubit of another.

Note, that the variables $Y_{\text{out,aux}}$ are auxiliar and not left open in the contraction. This amounts to not measuring them in a computational basis measurement.

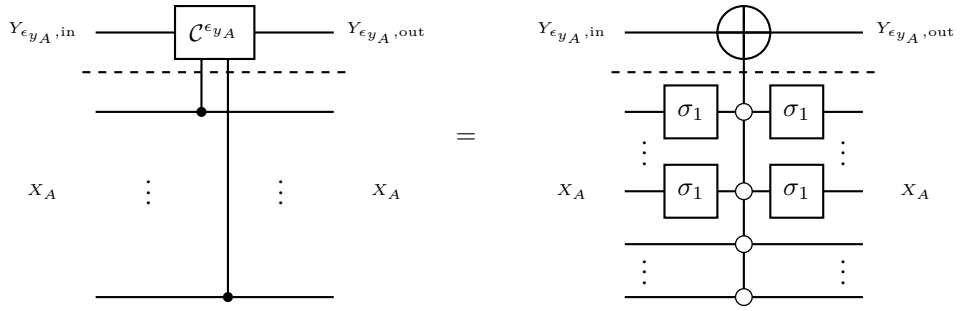


Figure 6: Exploitation of Polynomial sparsity in computation circuits. Here, we use the typical denotation of a muliple-controlled CNOT gate, which control symbols do not indicate Dirac tensors.

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

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

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.

Definition 4. *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 5. *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 6. *The computation circuit to a boolean tensor τ with a mod2-basis+ CP decomposition $\mathcal{M}$ obeys*

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

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 6. 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.

3.2.4 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:

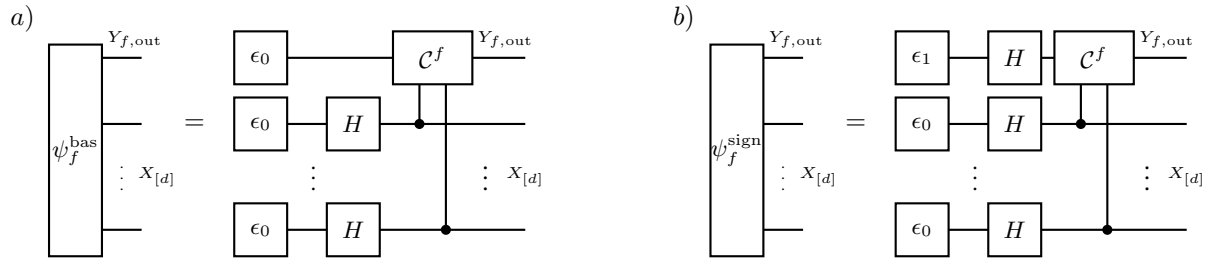


Figure 7: Construction of the basis encoding state (a) and the sign encoding state (b) using the computation circuit of a function.

- **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.

3.3 Preparation of function encoding states

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

3.3.1 Basis encoding

Definition 5. *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 7. *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 7a).

Corollary 1. *Using Lem. 3 we have*

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

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

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

Proof. We have

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

□

3.3.2 Sign encoding

Definition 6. *The sign encoding of a boolean function f is the state*

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

Lemma 9. *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 basis encoding state (see Figure 7b). Note that the initial state is a Hadamard gate acting on the state $\epsilon_1[Y]$.

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.

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

Lemma 10. *Let $f, \tilde{f}$ be two Boolean formulas on the states of $X_{[d]}$. Then we have*

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

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).

This encoding is applied in the Deutsch-Josza algorithm Deutsch and Jozsa (1992).

4 Computation of Contractions

We here study, how contractions of boolean formulas can be performed by circuits and read off by measurements. We develop a theory of *Quantum Parallelism* based on these contractions.

4.1 Deutsch-Josza Algorithm

We so far worked mainly with basis encoding states, which are prepared by a Hadamard transform and the computation circuit on statistic qubits in the ground state $\epsilon_0\left[Y\right]$. This ensured that the deterministic function relation is a hard constraint in the measurement distribution. If we apply a computation circuit on a different initial state of the statistic qubits, this constraint is in general not satisfied any more.

In the Deutsch-Josza algorithm Deutsch and Jozsa (1992), the Walsh-Hadamard transform of a sign encoding of a boolean function is measured to determine whether the function is constant. If f is constant, then the Walsh-Hadamard transform of the sign encoding results in the tensor product of the ground state $\epsilon_{0\dots0}\left[X_{[d]}\right]$ with a head qubit state.

Concatenating the computation circuit to formulas f and $\tilde{f}$ prepares the computation circuit of the formula $f \oplus \tilde{f}$. Both are equal or negations of each other, if and only if the distributed qubits are disentangled from the head qubit. Doing the Deutsch-Josza test on $f \oplus \tilde{f}$ thus decides whether two formulas are equal or negations of each other.

While the Deutsch-Josza Algorithm,

4.2 Inversion Test

The inversion test is checking whether $\psi^{\mathrm{bas}}_{f\oplus h}$ is (parallel to) uniform, using a Walsh-Hadamard transform. This is similar to the Deutsch-Josza algorithm, with the difference that the basis encoding instead of the sign encoding is used.

For two formulas f, h we have (see Lem. 8)

$$\left\langle \psi^{\mathrm{bas}}_f, \psi^{\mathrm{bas}}_h \right\rangle_{[\varnothing]} = \frac{1}{2^d} \cdot \left| \left\{ x_{[d]} \in \bigtimes_{k\in[d]} [2] \,:\, f\left[X_{[d]} = x_{[d]}\right] = h\left[X_{[d]} = x_{[d]}\right] \right\} \right| = \frac{1}{2^d} \cdot \left\langle f\left[X_{[d]}\right], h\left[X_{[d]}\right] \right\rangle_{[\varnothing]} .$$

Since we have by the computation circuit a circuit preparing the basis encoding states ψ^{bas}_f and ψ^{bas}_h we can apply the inversion test to investigate their overlap.

The inversion test is a quantum circuit, being a composition of the basis encoding circuits of two formulas f, h and their inverses, accompanied by Hadamard gates on all qubits before and after the circuit application. The probability of measuring the ground state $0, \dots, 0, 0$ in a computational basis measurement is then

$$\frac{1}{2^{2d}} \cdot \left\langle f\left[X_{[d]}\right], h\left[X_{[d]}\right] \right\rangle_{[\varnothing]} .$$

Further, the probability of measuring the state $0, \dots, 0, 1$ is

$$\frac{1}{2^{2d}} \cdot \left\langle f\left[X_{[d]}\right], \neg h\left[X_{[d]}\right] \right\rangle_{[\varnothing]} .$$

In such way, the contraction of two boolean formulas can be estimated by the rate of the ground state in a computational basis measurement.

4.3 Further Overlap-measuring Circuits

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?

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 Generic Q-samples

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

Definition 7 (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*

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

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). The q-sample can be prepared for Bayesian Networks, where each child qubit is prepared densely by C-NOTs conditioning on parent qubits.

Q-samples can be prepared by activation circuits acting on uniform quantum states (Hadamard gates acting on ground state).

Doing rejection sampling on the ancilla qubit corresponds with sampling from the normalized contraction with the activation tensor.

Lemma 11. *Given a distribution* $\mathbb{P}\left[X_{[d]}\right]$*, we construct a circuit preparing its q-sample and add the ancilla encoding of a tensor* $\tau\left[X_{[d]}\right]$*. The rejection sampling scheme, measuring the ancilla qubit and the* $X_{[d]}$ *qubits, rejecting the ancilla qubit measured as* 0*, prepares samples from the distribution*

$$\left\langle \mathbb{P}\left[X_{[d]}\right], \tau\left[X_{[d]}\right]\right\rangle_{\left[X_{[d]}|\varnothing\right]} .$$

5.2 Ancilla Augmentation

For more flexible sampling schemes of Computation-Activation Networks we need to introduce ancilla qubits.

Definition 8 (Ancilla Augmented Distribution)**.** *Let* $\mathbb{P}\left[X_{[d]}\right]$ *be a probability distribution over variables* $X_{[d]}$*. Another joint distribution* $\tilde{\mathbb{P}}$ *of* $X_{[d]}$ *and ancilla variables* $A_{[p]}$ *is called an ancilla augmented distribution, if*

$$\tilde{\mathbb{P}}[X_{[d]}|A_{[p]} = \mathbb{I}\left[[p]\right]] = \mathbb{P}\left[X_{[d]}\right] .$$

Sampling from the distribution can be done by rejection sampling on the ancilla augmented distribution, measuring all variables and rejecting all samples where an ancilla variable is 0.

Given an augmented Q-sample of a distribution, we can prepare samples from the distribution by rejection sampling, measuring all variables $X_{[d]}$ and $A_{[p]}$ and rejecting all samples where an ancilla qubit is measured as 0.

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.

Given a distribution $\mathbb{P}\left[X_{[d]}\right]$ we add an ancilla variable A and define the augmented distribution (see Figure 8)

$$\tilde{\mathbb{P}}[A, X_{[d]}] = \frac{1}{\prod_{k\in[d]} m_k} \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \epsilon_{x_{[d]}}\left[X_{[d]}\right] \otimes \left(\mathbb{P}\left[x_{[d]}\right] \cdot \epsilon_1\left[A\right] + \left(1 - \mathbb{P}\left[x_{[d]}\right]\right) \cdot \epsilon_0\left[A\right]\right) .$$

Then we have

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

5.3 Amplitude Amplification

Literature:

- Grover (1996) Grover algorithm (search in unstructured database)
- 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))

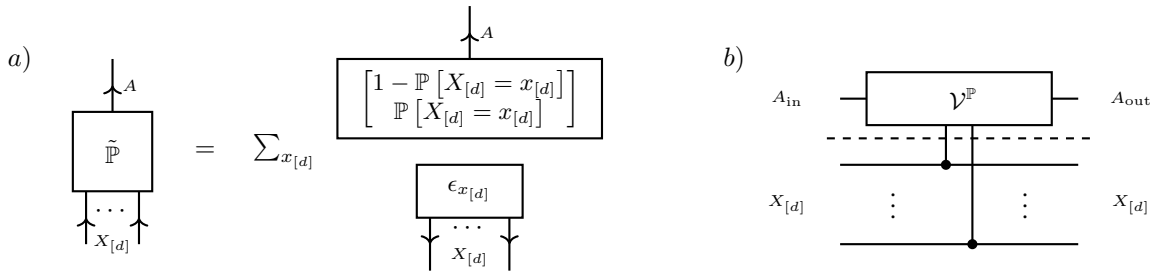


Figure 8: Ancilla augmentation of a distribution $\mathbb{P}\left[X_{[d]}\right]$. a) Augmented distribution $\tilde{\mathbb{P}}[A, X_{[d]}]$ with the property that $\mathbb{P}\left[X_{[d]}\right] = \tilde{\mathbb{P}}[X_{[d]}|A = 1]$. b) Preparation of the augmented distribution by the activation circuit of $\mathbb{P}\left[X_{[d]}\right]$.

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.

5.4 Sampling from Computation-Activation Networks as Quantum Circuits

So far: Sample from Hybrid Logic Networks, would need qudits for for more general Computation-Activation Networks.

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 3 (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

5.5 Acceptance Probability by ∞ Renyi Divergence

In more generality we draw samples from a generic proposal distribution $\mathbb{Q}$, which support needs to include the support of the target distribution $\mathbb{P}$. We draw a sample $x_{[d]}$ from $\mathbb{Q}$ and accept it with the probability

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

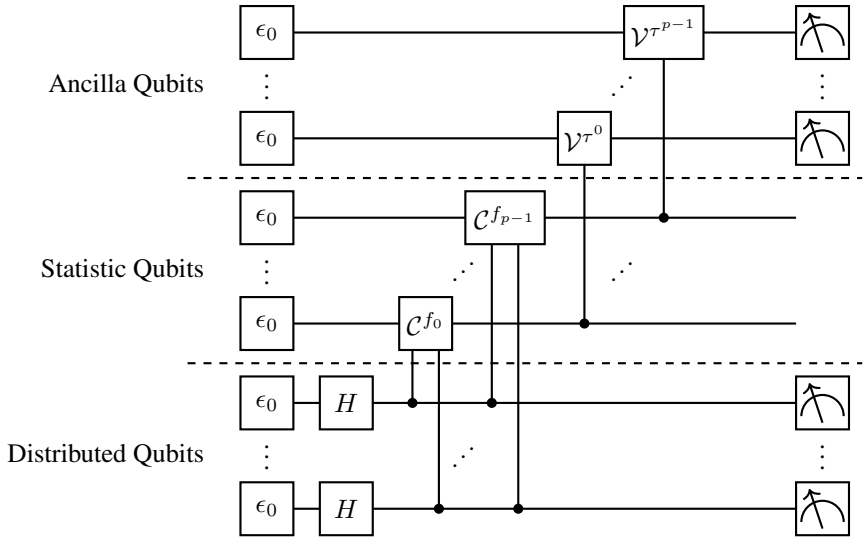


Figure 9: Quantum Circuit to reproduce a Computation-Activation Network (with elementary activation) by rejection sampling. We measure the distributed qubits $X_{[d]}$ and the ancilla qubits $A_{[p]}$ and reject all samples, where an ancilla qubit is measured as 0.

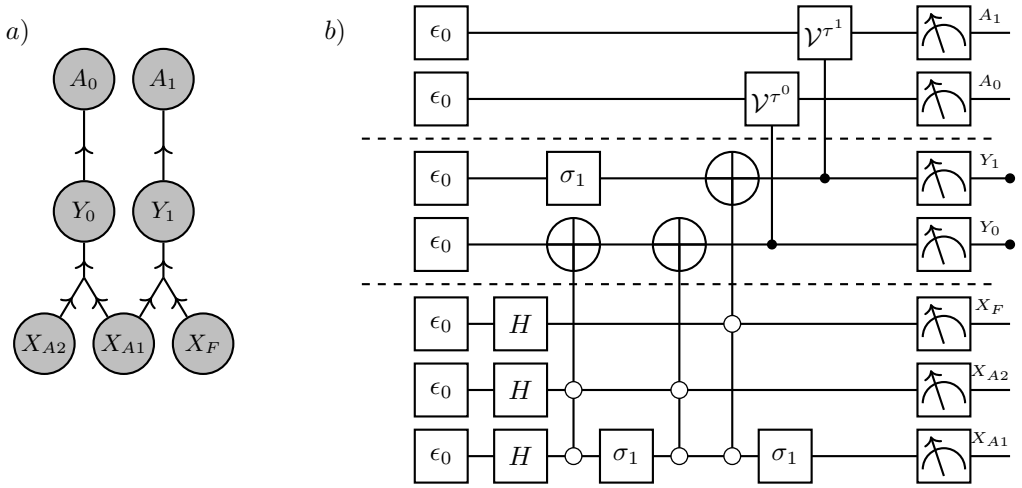


Figure 10: Circuit to sample from the Computation-Activation Network in the toy accounting Example 3.

The acceptance probability is the ∞ Renyi Divergence between the proposal distribution (typically uniform) and the target distribution.

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 $\alpha \to \infty$

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

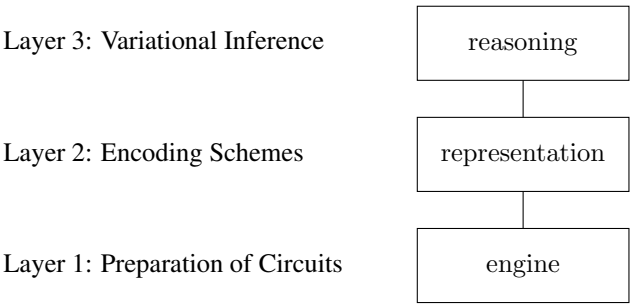
Then we have the acceptance probability

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

Extension: We could increase the acceptance probability, when we sample from a propsal 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.

6 Implementation

The introduced quantum circuit preparation schemes have been implemented in the python package qcreason, which consists in three layers:



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"*		$\sigma_1\left[A_{\text{in}}, A_{\text{out}}\right]$
Multiple controlled X	*"MCX"*		
Rotation around Y-axis	*"MRY"*	Angle α	$R_Y\left(\alpha\right)\left[A_{\text{in}}, A_{\text{out}}\right]$

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 angle 0.6121442808462428 is represented as:

```
[{'unitary': 'H',
    'targetQubits': ['blue']},
{'unitary': 'MCRY',
```

```
'targetQubits': ['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.

A 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 on a graph

Some constraints appear for a tensor network to be a quantum 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

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

- **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.
- **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.

B POVM measurements as contractions

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.

C 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

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

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

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

D Walsh-Hadamard transform

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

D.1 Transformation tensor

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

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

which coordinates are

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

We have

$$\langle \epsilon_0 [X_{\text{in}}], H [X_{\text{in}}, X_{\text{out}}] \rangle_{[X_{\text{out}}]} = \sqrt{\frac{1}{2}} \mathbb{I} [X_{\text{out}}]$$
$$\langle \epsilon_1 [X_{\text{in}}], H [X_{\text{in}}, X_{\text{out}}] \rangle_{[X_{\text{out}}]} = \sqrt{\frac{1}{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} [X_{\text{out}}]$$

and conversely

$$\left\langle \sqrt{\frac{1}{2}} \mathbb{I} [X_{\text{in}}], H [X_{\text{in}}, X_{\text{out}}] \right\rangle_{[X_{\text{out}}]} = \epsilon_0 [X_{\text{out}}]$$
$$\left\langle \sqrt{\frac{1}{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} [X_{\text{in}}], H [X_{\text{in}}, X_{\text{out}}] \right\rangle_{[X_{\text{out}}]} = \epsilon_1 [X_{\text{out}}] .$$

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/.

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.

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.

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.

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.