

# A Knowledge Compilation Map for Quantum Information

Lieuwe Vinkhuijzen,<sup>1</sup> Tim Coopmans,<sup>1,2</sup> Alfons Laarman<sup>1</sup>

<sup>1</sup>Leiden University, The Netherlands,

<sup>2</sup>QuTech, Delft University of Technology, The Netherlands

## Abstract

Despite their widespread use in quantum computing and physics, the relative strengths and weaknesses of Matrix Product States (MPS), Decision Diagrams (DDs), and Restricted Boltzmann Machines (RBMs) remains poorly understood. We analytically compare the succinctness of these quantum state representations and analyze the complexity of key operations on them. To overcome shortcomings of the tractability measure, we introduce ‘rapidity’ conditions that identify when non-canonical representations efficiently simulate each other. Our results reveal that: 1. Most DD variants are redundant with respect to MPS in a strong sense; MPS is more rapid. 2. Only one DD variant, called LIMDD, and RBM have succinctness incomparable to MPS. 3. LIMDD and RBM seem to achieve this by sacrificing tractability of counting queries, as shown by a metatheorem on the conditional hardness of these queries.

## 1 Introduction

Representing quantum information is hard as amplitude vectors grow exponentially, yet indispensable in many fields. Physicists use tensor networks (TNs; (Schollwöck 2011)), matrix product states (MPS; (Orús 2014)), restricted Boltzmann machines (RBMs; (Dumoulin et al. 2014; Carleo and Troyer 2017)) and the stabilizer formalism (Aaronson and Gottesman 2004). Computer scientists use decision diagrams (DDs; (Viamontes et al. 2003; Miller and Thornton 2006)). These methods have gradually converged, with hybrids of DDs and TNs (Hong et al. 2022; Burgholzer, Ploier, and Wille 2022), comparisons to probabilistic models (Glasser et al. 2019), and DDs extended with the stabilizer formalism (Vinkhuijzen et al. 2023).

However, the core differences between these representations remain elusive. Choosing one involves a trade-off between *succinctness* (representation size) and *tractability* (ease of operations like applying gates or measuring). Such trade-offs are well-understood in (probabilistic) reasoning (Darwiche and Marquis 2002; Fargier et al. 2014) and explainable AI (Audemard, Koriche, and Marquis 2020). Since a quantum system is but a small extension of a probabilistic system, a knowledge-representation perspective is also natural for quantum information, yet still lacking.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

In this work, we pioneer a knowledge compilation map for quantum state representations. We focus on representations developed for the above applications, formally defined in Sec. 3: algebraic DDs (ADD), semiring-labeled DDs (SLDD<sub>×</sub>, QMDD), local invertible map DDs (LIMDD), MPS and RBM. We select the relevant operations on these representations —such as, gate applications and computing measurements or fidelity— motivated by three applications: *simulating and verifying quantum circuits*, and *variational quantum algorithms*. *Variational methods* are the core of *quantum machine learning* (Benedetti et al. 2019; Dunjko and Briegel 2017), often used in quantum physics (Foulkes et al. 2001; Carleo and Troyer 2017). *Simulating quantum circuits* (Zulehner and Wille 2018; Thanos et al. 2023) is crucial for predicting performance under noise, thereby guiding hardware development. Finally, *verifying* if two quantum circuits are equivalent is crucial for checking if a (synthesized or optimized) circuit satisfies its specification (Ardehshir-Larijani, Gay, and Nagarajan 2014; Burgholzer and Wille 2020). For those, we study:

*Succinctness (Sec. 4)*: We find succinctness separations between MPS, RBM and LIMDD. We also find that MPS is strictly more succinct than SLDD<sub>×</sub>, contrary to earlier suggestion (Burgholzer, Ploier, and Wille 2022).

*Tractability (Sec. 5)*: We give a metatheorem showing that computing fidelity is conditionally intractable for representations that are succinct for both graph states and Dicke states. It applies to RBM and LIMDD. For SLDD<sub>×</sub> and LIMDD, we show that Hadamard and swap gates, might exponentially blow up the diagram.

*Rapidity (Sec. 6)*: Comparing tractability between representations  $D_1$  and  $D_2$  can deceive, because the asymptotic analysis uses different units:  $|D_1|$  versus  $|D_2|$ . To mend this deficiency, Lai, Liu, and Yin declared a representation  $D_1$  at least as rapid as  $D_2$  for operation  $OP$ , if  $D_1$  polynomially simulates (Groote and Zantema 2003)  $OP$  on  $D_2$  (taking  $|D_2|$  as a unit). We generalize the definition of rapidity for the non-canonical MPS and RBM. We then give a sufficient condition for a representation to be as rapid as another for all operations (satisfying reasonable criteria) and use it to settle several rapidity relations, showing surprisingly that MPS is strictly more rapid than SLDD<sub>×</sub> for the first time.

Here, we include proofs sketches, reserving full proofs for the report (Vinkhuijzen, Coopmans, and Laarman 2024).

## 2 Preliminaries: Quantum Information

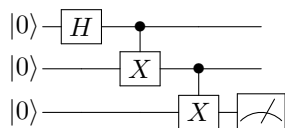
Quantum systems extend probabilistic systems by using the  $L_2$  instead of the  $L_1$  norm. We explain basics and refer to (Nielsen and Chuang 2000) for a complete introduction.

The basic unit of quantum information is a quantum bit or *qubit*. The joint state  $\varphi$  of  $n$  qubits is described by a *unit* vector of  $2^n$  complex numbers, denoted  $|\varphi\rangle \in \mathbb{C}^{2^n}$  in Dirac notation. Equivalently, a state  $\varphi$  is described by an *amplitude function*  $f_\varphi: \{0,1\}^n \rightarrow \mathbb{C}$  satisfying the norm  $\sum_{\vec{x} \in \{0,1\}^n} |f_\varphi(\vec{x})|^2 = 1$ . The computational basis of the  $n$ -qubit vector space  $\mathbb{C}^{2^n}$  is  $|k\rangle = (0, 0, \dots, 0, 1, 0, \dots, 0)^T$  for  $k \in \{0, 1, \dots, 2^n - 1\}$ , where the single entry 1 occurs at the  $k$ -th position. We often write  $k$  in binary:  $|k\rangle = |x_1 x_2 \dots x_n\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \dots \otimes |x_n\rangle$  for  $x_j \in \{0, 1\}$ . Thus  $|\varphi\rangle = \sum_{x \in \{0,1\}^n} f_\varphi(x) |x\rangle$ . Here  $A \otimes B$  is the tensor, or Kronecker, product. Two separate quantum states  $|\varphi_A\rangle, |\varphi_B\rangle$  on  $n_A, n_B$  qubits have joint state  $|\varphi_A\rangle \otimes |\varphi_B\rangle$ .

Two quantum states  $|\varphi\rangle, |\psi\rangle$  are *equivalent* iff there exists a complex value  $\lambda \in \mathbb{C}$  such that  $|\varphi\rangle = \lambda \cdot |\psi\rangle$ . Given two  $n$ -qubit states  $|\varphi\rangle, |\psi\rangle$ , their inner product is  $\langle \varphi | \psi \rangle \triangleq \sum_{\vec{x} \in \{0,1\}^n} f_\varphi(\vec{x})^* \cdot f_\psi(\vec{x})$ , and their fidelity is  $|\langle \varphi | \psi \rangle|^2$ . Note that, for  $k \in \{0, 1\}^n$ , the  $k$ -th entry in the state vector  $|\varphi\rangle$  is  $\langle k | \varphi \rangle = f_\varphi(k)$ .

A quantum state can be manipulated by applying a quantum *gate* or a measurement to it. An  $n$ -qubit gate  $U$  is a unitary transformation  $U: \mathbb{C}^{2^n} \rightarrow \mathbb{C}^{2^n}$ , i.e., it satisfies  $U^\dagger U = U U^\dagger = \mathbb{I}_{2^n}$  where  $\mathbb{I}_{2^n}$  is the  $n$ -qubit identity. Examples of gates are the four Pauli matrices defined in Sec. 3, the controlled- $U$  gate mapping  $|0\rangle \otimes |\varphi\rangle$  to itself and  $|1\rangle \otimes |\varphi\rangle$  to  $|1\rangle \otimes U|\varphi\rangle$ , the Hadamard gate  $H = 1/\sqrt{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ , the  $T = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{i} \end{bmatrix}$  and the phase gate  $S = T^2$ . A single-qubit computational-basis measurement, performed on the  $k$ -th qubit of an  $n$ -qubit state  $\varphi$ , yields outcome  $m \in \{0, 1\}$  with probability  $p_m \triangleq \sum_{x \in \{0,1\}^{n-1}} |f_\varphi(x, x_k = m)|^2$ , and the post-measurement state  $|\psi\rangle$  is found by setting a vector entries with  $x_k \neq m$  to 0 and subsequent rescaling; i.e.,  $f_\psi(x) := f_\varphi(x)/\sqrt{p_m}$  if  $x_k = m$  and  $f_\psi(x) := 0$  otherwise.

**Example 1.** A three-qubit quantum circuit is shown below. Its initial state is  $|0\rangle \otimes |0\rangle \otimes |0\rangle = |000\rangle$ . It first applies a Hadamard gate to the first qubit, yielding a state:  $1/\sqrt{2}(|000\rangle + |100\rangle)$ . Then a controlled- $X$  gate is applied to the first (control) and second qubit (target). The state of the quantum system is now  $1/\sqrt{2}(|000\rangle + |110\rangle)$ . The state before measurement is called the GHZ state:  $|\text{GHZ}\rangle = 1/\sqrt{2}(|000\rangle + |111\rangle)$ . Fig. 1 shows different representations of this state. Finally, a measurement is applied to the third qubit. After this measurement, the state “collapses” to either state  $|000\rangle$  or  $|111\rangle$  with equal probability  $\frac{1}{2} = |1/\sqrt{2}|^2$ .



## 3 Representations of Quantum States

In this section, we study representation languages (Def. 1), focussing on: Matrix Product States, Restricted Boltzmann Machines and Decision Diagrams. Fig. 1 shows examples.

**Definition 1.** [Inspired by Fargier et al.] A *quantum-state representation (language)* is a tuple  $(D, n, |\cdot\rangle, |\cdot|)$  where  $D$  is a set of representations. Given  $\alpha \in D$ ,  $|\alpha\rangle$  is the (possibly unnormalized) quantum state it represents (i.e., the interpretation of  $\alpha$ ),  $|\alpha|$  is the size of the representation, and  $n(\alpha)$ , or  $n$  in short, is the number of qubits of  $|\alpha\rangle$ . Finally, each quantum state should be expressible in the language.

We denote  $D^\varphi \triangleq \{\alpha \in D \mid |\alpha\rangle = |\varphi\rangle\}$ , i.e., the set of all instances in language  $D$  representing state  $|\varphi\rangle$ . By convention, we consider representations  $\alpha, \beta$  *equivalent* if  $|\alpha\rangle = \lambda |\beta\rangle$  for some  $\lambda \in \mathbb{C}$ .

**Matrix Product States (MPS).** An MPS  $M$  is a series of  $2n$  matrices  $A_k^x \in \mathbb{C}^{D_{k-1} \times D_k}$  where  $x \in \{0, 1\}$ ,  $k \in [n]$  and  $D_{k-1} \times D_k$  are the dimensions of the  $k$ -th matrix with  $D_0 = D_n = 1$ . The interpretation  $|M\rangle$  is determined as  $\langle \vec{x} | M \rangle = A_n^{x_n} \dots A_2^{x_2} A_1^{x_1}$  for  $\vec{x} \in \{0, 1\}^n$ . The size of  $M$  is the total number of matrix elements, i.e.,  $|M| = 2 \cdot \sum_{k=1}^n D_k \cdot D_{k-1}$ . By ‘the’ bond dimension of an MPS, we mean  $\max_{j \in [0..n]} D_j$ .

**Restricted Boltzmann Machine (RBM).** An  $n$ -qubit RBM is a tuple  $\mathcal{M} = (\vec{\alpha}, \vec{\beta}, W, m)$ , where  $\vec{\alpha} \in \mathbb{C}^n, \vec{\beta} \in \mathbb{C}^m$  for  $m \in \mathbb{N}_{\geq 1}$  are *bias vectors* and  $W \in \mathbb{C}^{n \times m}$  is a *weight matrix*. An RBM  $\mathcal{M}$  represents the state  $|\mathcal{M}\rangle$  as  $\langle \vec{x} | \mathcal{M} \rangle = e^{\vec{x}^T \cdot \vec{\alpha}} \cdot \prod_{j=1}^m (1 + e^{\beta_j + \vec{x}^T \cdot \vec{W}_j})$  where  $\vec{W}_j$  is column  $j$  of  $W$ ,  $\beta_j$  is entry  $j$  of  $\beta$  and where we write  $\vec{x}^T \cdot W_j$  to denote the inner product of the row vector  $\vec{x}^T$  and the column vector  $\vec{W}_j$  (Chen et al. 2018). The size of  $\mathcal{M}$  is  $|\mathcal{M}| = n + m + n \cdot m$ . We say this RBM has  $n$  *visible nodes* and  $m$  *hidden nodes*. A weight  $W_{v,j}$  is an edge from the  $v$ -th visible node to the  $j$ -th hidden node. The  $j$ -th hidden node is said to contribute the multiplicative term  $(1 + e^{\beta_j + \vec{x}^T \cdot \vec{W}_j})$ .

**Quantum Decision Diagrams (QDD).** Extending the Valued Decision Diagram from (Fargier et al. 2014), we define a QDD  $\alpha$  as a finite, rooted, directed acyclic graph  $(V, E)$ , where nodes  $v$  are labeled with qubit index  $\text{idx}(v) \in [n]$  and leaves have index 0. In addition, a ‘root edge’  $e_R$  (without a source node) points to the root node. Each node with label  $x$  has a ‘low edge’ (for  $x = |0\rangle$ ) and a ‘high edge’ (for  $x = |1\rangle$ ). Each edge  $e = vw$  pointing to node  $w$  with index  $k$  has a label  $\text{label}(e) \in \mathcal{E}_k$  for some edge label set  $\mathcal{E}_k$ , which is a group (for SLDD $_{\times}$  and LIMDD below with 0 added). Also, each leaf node  $v$  has a label  $\text{label}(v) \in \mathcal{L}$ . The size of a QDD is  $|\alpha| = |V| + |E| + \sum_{e \in E \cup \{e_R\}} |\text{label}(e)|$ . For simplicity, we require that no nodes are skipped, i.e.  $\forall vw \in E: \text{id}[v] = \text{id}[w] + 1$ .<sup>\*</sup> Hence, any node  $v$  at level  $k$ , i.e., with  $\text{idx}(v) = k$ , is  $k$  edges away from a leaf (along all paths) and therefore represents a  $k$ -qubit quantum state. The QDD semantics is:

<sup>\*</sup>The super-polynomial separations identified in our asymptotic analysis (Sec. 4, 5 and 6) are not affected as disallowing node skipping yields linear-size reductions at best (Knuth 2005).

(Quantum) decision diagrams (and variants)	Node merging strategy
Decision Tree	(no merging)
ADD (Bahar et al. 1997), MTBDD (Clarke et al. 1993), QuiDD (Viamontes et al. 2003)	$f = g$
SLDD $_{\times}$ (Wilson 2005; Fargier, Marquis, and Schmidt 2013) QMDD (Miller and Thornton 2006)	$f = p \cdot g$
LIMDD (Vinkhuijzen et al. 2023)	$f = pP_1 \otimes \dots \otimes P_n \cdot g$

Table 1: Various decision diagrams treated by the literature. The column *Merging strategy* lists the conditions under which two nodes  $v, w$ , representing subfunctions  $f, g: \{0, 1\}^k \rightarrow \mathbb{C}$  are merged. Here  $p, a \in \mathbb{C}$  are complex constants,  $P_i$  are Pauli gates and  $f + a$  means the function  $f(\vec{x}) + a$  for all  $\vec{x}$ .

- A leaf node  $v$  represents the value  $\text{label}(v)$ .
- A non-leaf node  $\bigcirc \xrightarrow{e_{low}} \dots \bigcirc \xrightarrow{e_{high}} \bigcirc$  represents  $|v\rangle = |0\rangle \otimes |e_{low}\rangle + |1\rangle \otimes |e_{high}\rangle$ .
- An edge  $\xrightarrow{e} \bigcirc$  represents  $|e\rangle = \text{label}(e) \cdot |v\rangle$ .

In this paper, we consider the following types of QDDs. We emphasize that an ADD can be seen as a special case of SLDD $_{\times}$ , which is a special case of LIMDD.

ADD:  $\forall k: \mathcal{E}_k = \{1\}, \mathcal{L} = \mathbb{C}$  (Bahar et al. 1997).

SLDD $_{\times}$ :  $\forall k: \mathcal{E}_k = \mathbb{C}, \mathcal{L} = \{1\}$  (Wilson 2005).

LIMDD:  $\forall k: \mathcal{E}_k = \text{PAULILIM}_k \cup \{0\}, \mathcal{L} = \{1\}$ , where  $\text{PAULILIM}_k$  is the Pauli group with complex weights (Vinkhuijzen et al. 2023), i.e.,  $\{\lambda P \mid \lambda \in \mathbb{C} \setminus \{0\}, P \in \{\mathbb{I}, X, Z, (-i)X \cdot Z\}^{\otimes k}\}$ .

$$\mathbb{I} \triangleq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X \triangleq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Z \triangleq \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Two isomorphic QDD nodes (Def. 2)  $v, w$  with  $|w\rangle = \ell|v\rangle$  can be *merged* by removing  $w$  and rerouting all edges  $uw \in E$  incoming to  $w$  to  $v$ , updating their edge labels accordingly (i.e.,  $\text{label}(uw) := \text{label}(uv) \cdot \ell$ ). Table 1 summarizes merging strategies for various QDDs. If all isomorphic nodes are merged, we call a QDD *reduced*. We may assume a QDD is reduced (and even canonical), since reduction takes polynomial time and is done on-the-fly in manipulation algorithms (Bryant 1986; Miller and Thornton 2006; Fargier, Marquis, and Schmidt 2013; Vinkhuijzen et al. 2023).

**Definition 2** (Isomorphic nodes). *QDD node  $v$  is isomorphic to node  $w$ , if there exists an edge label  $\ell \in \mathcal{E}_{\text{idX}(v)}, \ell \neq 0$*

such that  $\ell \cdot |v\rangle = |w\rangle$ . Since  $\mathcal{E}_{\text{idX}(v)}$  is a group, isomorphism is an equivalence relation.

We present additional background required to understand the results on decision diagrams. See (Somenzi 1999) for a detailed treatment.

It is well known that the variable order greatly influences QDD sizes (Bollig and Wegener 1996; Darwiche 2011; Bova 2016). Our results here assume any variable order. So if a structure  $D$  is strictly more succinct than a QDD, then there is no variable order for which the QDD is more succinct than  $D$  (for representing a certain worst-case state).

The tractability of the QDD algorithms heavily relies on two properties: canonicity, i.e., there exists a unique decision diagram for each quantum state, and dynamic programming, i.e., avoiding unnecessary recursion by storing intermediary results in a cache. Canonicity means that for each quantum state  $\varphi$  there is a unique diagram  $x \in D^{\varphi}$ , and moreover no other (non-canonical) representation  $y \in D^{\varphi}$  has fewer nodes or edges than  $x$ . All QDDs in this work can be made canonical by requiring that the QDD satisfies a set of reduction rules (for SLDD $_{\times}$  see (Miller and Thornton 2006) and for LIMDD see (Vinkhuijzen et al. 2023)) and all isomorphic nodes are subsequently merged (see Table 1). This so-called *reduced* QDD can be obtained in polynomial time using a standard `MAKENODE` procedure (see also Sec. 6.2). QDD manipulation algorithms also use the `MAKENODE` procedure to ensure that the result is reduced efficiently. For these reasons, we may always assume that QDDs are reduced.

Let  $f: \{0, 1\}^n \rightarrow \mathbb{C}$  be a function, and  $y_n, y_{n-1}, \dots, y_k \in \{0, 1\}$  a partial assignment. Then  $f_y: \{0, 1\}^{k-1} \rightarrow \mathbb{C}$  is the *subfunction of  $f$  induced by  $y$* , defined as  $f_y(x) = f(y, x)$ . Suppose a QDD has root node

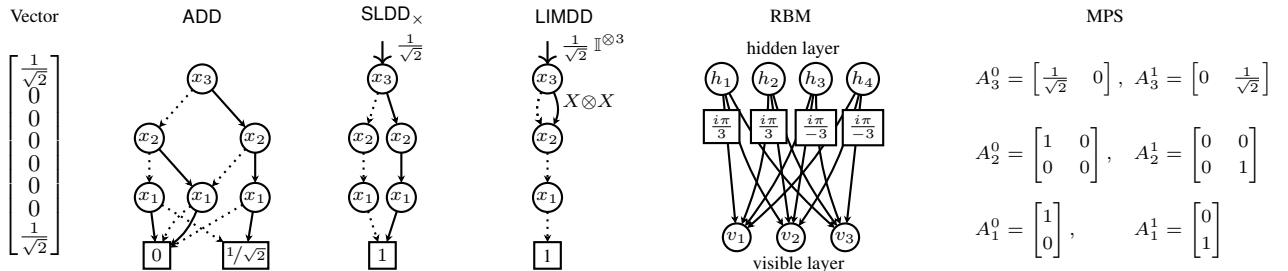


Figure 1: The three-qubit GHZ state  $1/\sqrt{2}(|000\rangle + |111\rangle)$ , in six representations. Dashed (solid) QDD edges are  $|0\rangle$  ( $|1\rangle$ ) edges. The unlabelled QDD edges have label 1 ( $\mathbb{I}^{\otimes k}$ ), while omitted edges have weight 0. In the RBM, the weights of edges incident to  $h_1, h_2$  ( $h_3, h_4$ ) are all  $i\pi/3$  ( $-i\pi/3$ ); the hidden node biases  $(\beta_{h_1}, \beta_{h_2}, \beta_{h_3}, \beta_{h_4}) = i\pi \cdot (1/3, 2/3, -1/3, -2/3)$ ; the visible node biases  $\alpha_{v_1} = \alpha_{v_2} = \alpha_{v_3} = 0$ .

$v$  representing a pseudo-Boolean function  $f: \{0, 1\}^n \rightarrow \mathbb{C}$ . For any  $y_n, \dots, y_k$ , this diagram contains a node  $w$  which represents the induced subfunction  $f_y: \{0, 1\}^{k-1} \rightarrow \mathbb{C}$ , or which represents a function which is isomorphic to  $f_y$  under  $\mathcal{E}_k$ . The vertices of the QDD are divided into *layers*, i.e., a node which is  $k$  edges away from a leaf is said to be in layer  $k$ , and each edge points from a vertex in layer  $k$  to a node in layer  $k - 1$ . Each node in layer  $k$  represents a pseudo-Boolean function of the form  $f: \{0, 1\}^k \rightarrow \mathbb{C}$ .

Prop. 1 formalizes intuition used in some of our proofs.

**Proposition 1.** *A reduced QDD representing pseudo-Boolean function (quantum state)  $f$  has as many nodes on level  $l$ , as there are unique induced subfunctions  $f_{\vec{a}}$  for  $\vec{a} \in \{0, 1\}^{n-l}$ , modulo isomorphism under  $\mathcal{E}_k$  (see Def. 2).*

## 4 Succinctness of Representations

A class of representations  $D_1$  is as succinct as  $D_2$  if  $D_2$  uses at least as much space to represent any quantum state as  $D_1$ , up to polynomial factors. Thus, a more succinct diagram is more expressive when we constrain ourselves to polynomial (or asymptotically less) space. We define a representation  $D_1$  to be *at least as succinct as*  $D_2$  (written  $D_1 \preceq_s D_2$ ), if there exists a polynomial  $p$  such that for all  $\beta \in D_2$ , there exists  $\alpha \in D_1$  such that  $|\alpha\rangle = |\beta\rangle$  and  $|\alpha| \leq p(|\beta|)$ . We say that  $D_1$  is (strictly) more succinct ( $D_1 \prec_s D_2$ ) if  $D_1$  is at least as succinct as  $D_2$  but not vice versa.

The results of this section are summarized by Th. 1 (Fig. 2). We highlight our novel results in blue in Fig. 2.

**Theorem 1.** *The succinctness results in Fig. 2 hold.*

*Proof outline of Th. 1.* We consider edges A,B,C in Fig. 2.

A: We shall see in the proof sketch of Lem. 2 that any  $\text{SLDD}_\times$  (and ADD) can be transformed into an MPS with bond matrices that contain only one non-zero entry per row (reflecting the DD's determinism). Interestingly, as MPS does not need to adhere to this constraint, an MPS can represent functions that are hard for  $\text{SLDD}_\times$  with only bond dimension two; e.g., the  $|\text{Ham}^n\rangle$  state:

$$|\text{Ham}^n\rangle = \sum_{\vec{x} \in \{0,1\}^n} \left( \sum_{j=1}^n 2^{j-1} \vec{x}_j \right) |\vec{x}\rangle, \text{ which has MPS:}$$

$$\begin{aligned} A_n^0 &= [1 \ 0] & A_j^0 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & A_1^0 &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ A_n^1 &= [1 \ 2^{n-1}] & A_j^1 &= \begin{bmatrix} 1 & 2^{j-1} \\ 0 & 1 \end{bmatrix} & A_1^1 &= \begin{bmatrix} 1 \end{bmatrix} \end{aligned} \quad (1)$$

The following two states are exponentially-sized in the representation at the edge marked by resp. B,C, while having provably polynomial size in the other representation at the opposite edge end.

$$\text{B: } |\text{Sum}^n\rangle = (|0\rangle + |1\rangle)^{\otimes n} + \bigotimes_{j=1}^n (|0\rangle + e^{i\pi 2^{-j-1}} |1\rangle)$$

$$\text{C: } |\text{IP}^n\rangle = \sum_{\vec{x} \in \{0,1\}^{2n}} \left( \sum_{j=1}^n \vec{x}_{2j} \vec{x}_{2j+1} \pmod 2 \right) |\vec{x}\rangle \quad \square$$

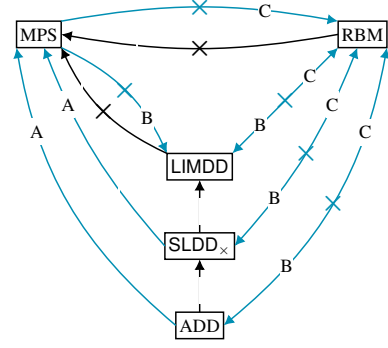


Figure 2: Succinctness relations between various representations of quantum states. Solid arrows  $D_1 \rightarrow D_2$  denote  $D_2 \preceq_s D_1$ , i.e.,  $D_2$  is strictly more succinct than  $D_1$ . Crossed arrows  $D_1 \not\rightarrow D_2$  denote a separation  $D_2 \not\preceq_s D_1$ ; a bidirectional crossed arrow implies incomparability. Blue arrows indicate novel relations that we identified.

## 5 Tractability of Quantum Operations

By a *manipulation operation*, we mean a map  $D^c \rightarrow D$  and by a *query operation* a map  $D^c \rightarrow \mathbb{C}$ , where  $D$  is a class of representations and  $c \in \mathbb{N}_{\geq 1}$  the number of operands. We say that a class of representations  $D$  *supports* a (query or manipulation) operation  $OP(D)$ , if there exists an algorithm implementing  $OP$  whose runtime is polynomial in the size of the operands, i.e.,  $|\varphi_1| + \dots + |\varphi_c|$ . The operations whose tractability we investigate are listed below and follow from the following formalization of the three applications domains introduced in Sec. 1.

The first domain is *circuit simulation*. A quantum circuit consists of (i) an initial quantum state, (ii) several quantum gates followed by (iii) measuring one or more quantum bits. Simulating a circuit using the representations in this work starts by constructing a representation for (i), followed by manipulating the representation by applying the gates and measurements of (ii) and (iii) one by one. The structure supports *strong simulation* if it can produce the probability function; or *weak simulation* if it merely produces a sample, for each measurement in (iii). Next, *variational methods* start with a quantum circuit; then the output state  $|\varphi\rangle$  is used to compute  $\langle \varphi | O | \varphi \rangle$  for some linear operator  $O$  (an *observable*) which is hermitian ( $O = O^\dagger$ ), for example  $O$  describes the energy of the quantum system, in which case it is called the Hamiltonian. This computation reduces to  $\langle \varphi | \psi \rangle$  with  $|\psi\rangle := O |\varphi\rangle$  which could be any complex vector. Often one considers *local* observables. Specifically, an  $n$ -qubit observable  $O$  is called  $k$ -local when it can be written as  $O = P \otimes \mathbb{I}_{2^{n-k}}$  where  $P$  is a  $k$ -qubit observable. Last, *circuit verification* relies on checking approximate or exact equivalence of quantum states. This extends to unitaries, which all representations from this paper can represent as well, but which we will not treat for simplicity. For any representation  $D$ , we consider:

- **Sample:** Given  $|\varphi\rangle$  in  $D$ , sample outcomes  $\vec{x} \in \{0, 1\}^n$  from measuring all qubits of  $|\varphi\rangle$  in computational basis.
- **Measure:** Given  $|\varphi\rangle$  in  $D$  and  $\vec{x} \in \{0, 1\}^n$ , compute the probability to obtain outcome  $\vec{x}$  when measuring  $|\varphi\rangle$ .

	Queries					Manipulation operations						
	Sample	Measure	Equal	InnerProd	Fidelity	Addition	Hadamard	X,Y,Z	CZ	Swap	Local	T-gate
Vector	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ADD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SLDD <sub>×</sub>	✓	✓	✓	✓	✓	✘	✘	✓	✓	✘	✘	✓
LIMDD	✓	✓	✓	○	○	✘	✘	✓	✓	✘	✘	✓
MPS	✓	✓	✓	○	○	?	✓	✓	✓	✓	✓	✓
RBM	✓	?	?	○	○	?	?	✓	✓	✓	?	✓

Table 2: Tractability of queries and manipulation operations. A ✓ (✓') means that the representation supports the operation in (randomized) polytime. A ✘ (○) means it does not (unless  $P = NP$ ). ? means unknown. The table only considers deterministic algorithms (for e.g. **InnerProd** on RBM, a probabilistic algorithm exists). Novel results are blue and underlined.

- **Equal** (Equality): For  $|\varphi\rangle, |\psi\rangle$  in  $D$ , decide equivalence.
- **InnerProd** (inner product) and **Fidelity**: Given  $|\psi\rangle, |\psi\rangle$  in  $D$ , compute  $\langle\varphi|\psi\rangle$  and  $|\langle\varphi|\psi\rangle|^2$ .
- **Addition**: Given DSs for states  $|\varphi\rangle, |\psi\rangle$ , construct a  $D$ -representation of state  $|\varphi\rangle + |\psi\rangle$ .
- **Gates**: Given  $|\varphi\rangle$  in  $D$  and a gate  $U$  from the universal set  $\{\text{Hadamard, X,Y,Z, CZ, Swap, T}\}$  (Bravyi and Kitaev 2005), construct a  $D$ -representation of  $U|\varphi\rangle$ .
- **Local**: Given  $|\varphi\rangle$  in  $D$ , a constant  $k \in \mathbb{N}_{\geq 1}$  and a  $k$ -local gate  $U$ , construct a  $D$ -representation of the state  $U \cdot |\varphi\rangle$ . By  $k$ -local, we mean acting on some subset of  $k$  qubits.

Although **Addition** is not, technically speaking, a quantum operation, its inclusion is instructive because addition is part of some quantum operations. For example, if we apply first a Hadamard gate, and then a measurement, to the state  $|0\rangle|\varphi\rangle + |1\rangle|\psi\rangle$ , we may obtain the state  $|0\rangle \otimes (|\varphi\rangle + |\psi\rangle)$ .

For the above operations and the representations from Sec. 3, we present an overview of tractability results in Th. 2 (Table 2). The novel results in this work are the hardness results (denoted ○) for **InnerProd** and **Fidelity** on LIMDD and RBM (Corollary 1 and for **InnerProd** we can reduce from **Fidelity**) and unsupported manipulations (denoted ✘) on SLDD<sub>×</sub> and LIMDD. More generally, our proof shows that computing **Fidelity** is hard for any representation which can efficiently construct (and thus also succinctly represent) both all graph states and all Dicke states (Th. 3). Both RBM and LIMDD fall in this category. The proof uses a reduction from the #EVEN SUBGRAPHS problem (Jerrum and Meeks 2017). Also novel is the result that fidelity is supported by SLDD<sub>×</sub>: A fidelity algorithm for SLDD<sub>×</sub> was mentioned by (Burgholzer, Kueng, and Wille 2021), but had not previously been described or analyzed.

**Theorem 2.** *The tractability results in Table 2 hold.*

In the following metatheorem, we use graph states  $|G\rangle$  (Van den Nest, Dehaene, and De Moor 2004) and Dicke states  $|D_n^k\rangle$  (Dicke 1954; Bärttschi and Eidenbenz 2019). Given an undirected graph  $G = (V, E)$  with  $V = [1..n]$

and a Hamming weight  $k \leq n$ , these are defined as follows, where  $|G[\vec{x}]|$  for  $\vec{x} \in \{0, 1\}^n$  represents the number of edges in  $G$ 's subgraph on vertices  $\{j \in [1..n] \mid x_j = 1\} \subseteq V$ .

$$|G\rangle = \frac{1}{\sqrt{2^n}} \sum_{\vec{x} \in \{0,1\}^n} (-1)^{|G[\vec{x}]|} |\vec{x}\rangle, \quad (2)$$

$$|D_n^k\rangle = \frac{1}{\sqrt{\binom{n}{k}}} \sum_{\vec{x} \in \{0,1\}^n \text{ with } |\vec{x}|=k} |\vec{x}\rangle, \quad (3)$$

**Theorem 3.** *Let  $D$  be a representation which has a poly( $n$ )-sized, efficiently constructable representation of each Dicke state and each graph state ( $n$  is the number of qubits). Assuming the exponential time hypothesis, the first  $\Omega(n)$  bits of the fidelity between two states represented in  $D$  generally cannot be computed in polytime in  $n$ .*

*Proof sketch.* We prove hardness of inner product first, by reducing inner product from the #EVEN-SUBGRAPHS problem, which asks how many induced subgraphs of a given graph have an even number of edges and which is hard for the complexity class #W[1] (Jerrum and Meeks 2017) hence cannot be solved in polynomial time, unless the exponential time hypothesis (ETH) is false. Formally, given as input a graph  $G = (V, E)$  on  $n$  vertices  $V$  with edge set  $E$ , and  $0 \leq k \leq n$  an integer; the goal is to find  $e(G, k)$ , the number of induced subgraphs  $G[\vec{x}]$  on  $|\vec{x}| = k$  vertices  $\{j \mid x_j = 1\} \subseteq V$  which have a number of edges  $|G[\vec{x}]|$  that is even. For the reduction, we note that:

$$e(G, k) = \frac{1}{2}|I| + \frac{1}{2}\sqrt{\binom{n}{k} 2^n} \langle D_n^k | G \rangle \quad (4)$$

where  $|I| = \binom{n}{k}$  denotes the total number of  $k$ -induced subgraphs of  $G$ . The reduction thus first constructs poly-size representations for  $|G\rangle$  and  $|D_n^k\rangle$ , then computes the fractional part of  $\langle D_n^k | G \rangle \in [0, 1)$  up to the first  $\Omega(\log_2(\frac{1}{2}\sqrt{\binom{n}{k} 2^n})) \leq \Omega(\log_2(\frac{1}{2}\sqrt{2^n \cdot 2^n})) = \Omega(n)$  bits. This number of bits suffices to determine the half-integer  $\frac{1}{2}\sqrt{\binom{n}{k} 2^n} \langle D_n^k | G \rangle$ , from which we find the integer  $e(G, k)$  using Eq. (4).

Since fidelity reduces to inner product, proving its hardness requires an extension of the above construction. We achieve this in the technical report (Vinkhuijzen, Coopmans, and Laarman 2024) by defining a new EVEN ODD SUBGRAPHS DIFFERENCE (EOSD) problem, and providing two reductions: from #EVEN SUBGRAPHS to EOSD, and from EOSD to FIDELITY.  $\square$

**Corollary 1.** *Assuming the exponential time hypothesis holds, the fidelity of two states cannot be efficiently computed in LIMDD nor in RBM.*

## 6 Rapidity of Representations

The tractability criterion, studied in Sec. 5, sometimes gives a skewed picture of efficiency because a different unit is used for each representation (its size). Looking naively at Table 2,

it seems for instance that  $\text{ADD}$  is faster than  $\text{SLDD}_\times$  when applying a Hadamard gate. Yet there is no state for which the  $\text{SLDD}_\times$  representation is slower than the  $\text{ADD}$  representation for this operation. Instead, the correct interpretation of this result is that the  $\text{SLDD}_\times$  is exponentially more succinct than  $\text{ADD}$ , but applying the Hadamard gate sometimes yields an output state whose  $\text{SLDD}_\times$  is as large as its  $\text{ADD}$ ; however, tractability measures runtime in terms of the size of the input state's diagram. To remedy this shortcoming, Lai, Liu, and Yin (Lai, Liu, and Yin 2017) introduced the notion of *rapidity* for canonical representations.

In Def. 3, we generalize rapidity to support non-canonical representations, such as MPS, RBM, d-DNNF (Darwiche 2001) and CCDD (Lai, Meel, and Yap 2022). To achieve this, Def. 3 requires that for a fixed input  $\varphi$  to  $\text{ALG}_1$ , among all equivalent inputs to  $\text{ALG}_2$ , there is one on which  $\text{ALG}_2$  is at least as fast as  $\text{ALG}_1$ . It may seem reasonable to require, instead, that  $\text{ALG}_2$  is at least as fast as  $\text{ALG}_1$  on all equivalent inputs; however, in general, there may be no upper bound on the size of the (infinitely many) such inputs; thus, such a requirement would always be vacuously false.

**Definition 3.** Let  $D_1, D_2$  be two representations and  $OP$  a  $c$ -ary operation. In the below,  $\text{ALG}_1$  ( $\text{ALG}_2$ ) is an algorithm implementing  $OP$  for  $D_1$  ( $D_2$ ). We will write  $\text{time}(A, x)$  for the runtime of algorithm  $A$  on input  $x$ .

- (a) We say that  $\text{ALG}_1$  is at most as rapid as  $\text{ALG}_2$  (or  $\text{ALG}_2$  is at least as rapid as  $\text{ALG}_1$ ) iff there exists a polynomial  $p$  such that for each input  $\varphi = (\varphi_1, \dots, \varphi_c)$  there exists an equivalent input  $\psi = (\psi_1, \dots, \psi_c)$ , i.e., with  $|\varphi_j\rangle = |\psi_j\rangle$  for  $j = 1 \dots c$ , for which  $\text{time}(\text{ALG}_2, \psi) \leq p(\text{time}(\text{ALG}_1, \varphi))$ .
- (b) We say that  $OP(D_1)$  is at most as rapid as  $OP(D_2)$  if for each algorithm  $\text{ALG}_1$  performing  $OP(D_1)$ , there is an algorithm  $\text{ALG}_2$  performing  $OP(D_2)$  which is at most as rapid as  $\text{ALG}_2$ .

We remark that, when applied to canonical representations, Def. 3 reduces to the definition by Lai, Liu, and Yin, except that Lai et al. allow the input to be fully read by the algorithm:  $\text{time}(\text{ALG}_1, x_1) \leq \text{poly}(\text{time}(\text{ALG}_2, x_2) + |x_2|)$  (difference underlined). We omit this to achieve transitivity (Lai, Liu, and Yin's definition loses transitivity for algorithms which are faster than the time it takes to read the entire input, as then the  $|x_2|$  term dominates). Rapidity indeed has the desirable property that it is a preorder, i.e., it is reflexive (in the rapidity definition, Def. 3, choose  $\psi = \varphi$  and for  $p$ , choose the identity polynomial  $x \mapsto x$ ) and transitive (follows from the fact that the composition of polynomials is again a polynomial).

**Theorem 4.** Rapidity is a preorder over representations.

### 6.1 A Sufficient Condition for Rapidity

In Th. 5, we introduce a simple sufficient condition for rapidity, allowing researchers to easily establish that one representation is more rapid than another *for many relevant operations simultaneously*. Previously, such proofs were done for each operation individually (Lai, Liu, and Yin 2017). We use this sufficient condition to establish rapidity relations between many of the representations studied in this work. By a

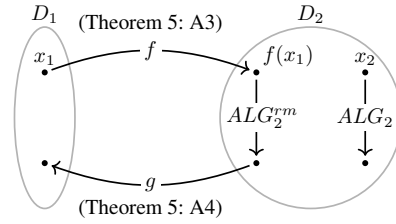
transformation  $f$  from representation  $D_1$  to  $D_2$  we mean a map such that  $|f(x_1)\rangle = |x_1\rangle$  for all  $x_1 \in D_1$ . We also need the notions of a weakly minimizing transformation (Def. 4) and a runtime monotonic algorithm (Def. 5).

**Definition 4** (Weakly minimizing transformation). Let  $D_1, D_2$  be representations. A transformation  $f : D_1 \rightarrow D_2$  is weakly minimizing if  $f$  always outputs an instance which is polynomially close to minimum-size, i.e., there exists a polynomial  $p$  such that for all  $x_1 \in D_1, x_2 \in D_2$  with  $|x_1\rangle = |x_2\rangle$ , we have  $|f(x_1)| \leq p(|x_2|)$ .

**Definition 5** (Runtime monotonic algorithm). An algorithm  $\text{ALG}$  implementing some operation on representation  $D$  is runtime monotonic if for each polynomial  $s$  there is a polynomial  $t$  such that for each state  $|\varphi\rangle$  and each  $x, y \in D^\varphi$ , if  $|x| \leq s(|y|)$ , then  $\text{time}(\text{ALG}, x) \leq t(\text{time}(\text{ALG}, y))$ .

**Theorem 5.** Let  $D_1, D_2$  be representations with  $D_1 \preceq_s D_2$  and  $OP$  a  $c$ -ary operation. If conditions A1-A4 hold, then  $D_1$  is at least as rapid as  $D_2$  for operation  $OP$ .

- A1  $OP(D_2)$  requires time  $\Omega(m)$  where  $m$  is the sum of the sizes of the operands.
- A2 For each algorithm  $\text{ALG}$  implementing  $OP(D_2)$ , there is a runtime monotonic algorithm  $\text{ALG}^{rm}$ , implementing  $OP(D_2)$ , which is at least as rapid as  $\text{ALG}$ .
- A3 There exists a transformation from  $D_1$  to  $D_2$  which is (i) weakly minimizing and (ii) runs in time  $\text{poly}(|\varphi| \cdot |\psi|)$  for transformation input  $\varphi \in D_1$  and output  $\psi \in D_2$ .
- A4 If  $OP$  is a manipulation operation (i.e. not a query), then there exists a polytime transformation from  $D_2$  to  $D_1$  (i.e. in time  $\text{poly}(|\rho|)$  for input  $\rho \in D_2$ ).



*Proof sketch.* The figure above visualizes the case when  $OP$  is a manipulation operation: given transformations  $f, g$  satisfying items A3 and A4 and a runtime monotonic algorithm  $\text{ALG}_2^{rm}$  implementing  $OP$  on language  $D_2$  (Item A2), the composed algorithm  $\text{ALG}_1 \triangleq g \circ \text{ALG}_2^{rm} \circ f$  for  $OP(D_1)$  is at least as rapid as  $\text{ALG}_2$ . A1 is needed to ensure  $f, g$  have time to finish converting representations, e.g. if  $\text{ALG}_2$  is has  $O(1)$  runtime, then no transformations  $f$  that makes  $g \circ \text{ALG}_2^{rm} \circ f$  more rapid, exists, as  $f$  would not even have the time to read the entire input.  $\square$

We opted for weakly minimizing transformations (rather than *strictly* minimizing transformations), because a minimum structure might be hard to compute and is not needed in the proof. Runtime monotonic algorithms are ubiquitous, for instance, most operations on MPS scale polynomially in the bond dimension and number of qubits (Vidal 2003). Finally, we emphasize that for canonical representations  $D$ , each algorithm is runtime monotonic and any transformation  $D_1 \rightarrow D$  is weakly minimizing.

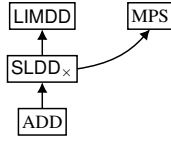


Figure 3: The rapidity relations identified in this work. An arrow  $D_1 \rightarrow D_2$  means  $D_2$  is at least as rapid as  $D_1$  for operations satisfying A1 and A2 of Th. 5.

## 6.2 Rapidity Relations Between Representations

We now capitalize on the sufficient condition of Th. 5 by revealing the rapidity relations between representations for all operations satisfying A1 and A2. Th. 6 shows our findings. We highlight the result that MPS is at least as rapid as  $SLDD_x$  in Corollary 2, which we prove by providing the required transformations from MPS to  $SLDD_x$  and back (Lem. 1 and Lem. 2).

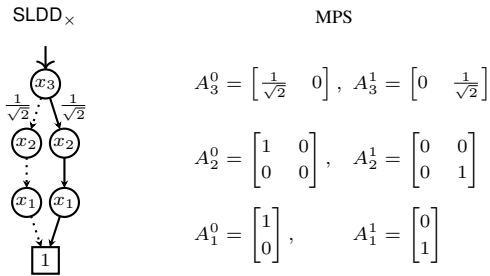
**Theorem 6.** *The rapidity relations in Fig. 3 hold.*

**Lemma 1.** *There is a weakly minimizing transformation from MPS to  $SLDD_x$ , that runs in time polynomial in the product of the sizes of the MPS and the resulting  $SLDD_x$ .*

*Proof sketch.* We provide a recursive algorithm which constructs the  $SLDD_x$  bottom-up: given an MPS  $\{A_n^0, A_n^1, A_{n-1}^0, A_{n-1}^1\} \cup A$  with  $A = \{A_{n-2}^0, A_{n-2}^1, \dots, A_1^0, A_1^1\}$  for state  $|\varphi\rangle = |0\rangle|\varphi_0\rangle + |1\rangle|\varphi_1\rangle$ , the state  $|\varphi_0\rangle$  is represented by  $\{A_n^0 \cdot A_{n-1}^0, A_n^0 \cdot A_{n-1}^1\} \cup A$ , and similarly for  $|\varphi_1\rangle$ . We use dynamic programming: to an MPS whose  $SLDD_x$  node we have already constructed, we return an edge to that  $SLDD_x$  node without recursing further. We identify such *equivalent* cases by checking if the inner product with an existing node is 1, and inner product for MPS is tractable.  $\square$

**Lemma 2.** *There exists a polynomial-time transformation which converts an  $SLDD_x$  to an equivalent MPS.*

*Proof sketch.* We let the  $A_\ell^0$ -matrix of the MPS represent the adjacency matrix of the  $|0\rangle$ -edges from level  $\ell$  to  $\ell + 1$  of the  $SLDD_x$ . Similarly, the  $A_\ell^1$ -matrices represent the  $|1\rangle$ -edges. The figures below illustrate the transformation from  $SLDD_x$  to MPS with an example modifying the  $SLDD_x$  from Fig. 1 by ‘pushing down’ the root label. The representations still captures the same state.



From graph theory, we know that for any  $\vec{x} \in \{0, 1\}^n$  multiplying weighted adjacency matrices  $A_n^{x_n} \cdots A_1^{x_1}$  yields

the sum of products of weights along all length- $n$  paths from root to leaf. By the determinism of the  $SLDD_x$  this yields a single amplitude as there is only a single  $\vec{x}$ -path.  $\square$

**Corollary 2.** *MPS is at least as rapid as  $SLDD_x$  for all operations satisfying A1 and A2.*

The relations not involving MPS involve QDDs. QDDs are canonical representations as explained in Sec. 3, so that runtime monotonicity and weak minimization of transformations are automatically ensured. It is a standard procedure to transform different QDDs using the well-known `MAKENODE` procedures (to construct `ADD` or `SLDD_x`; `MAKEEDGE` to construct `LIMDD`), in polynomial time in the resulting QDD size (using dynamic programming): see the discussion on canonicity in Sec. 3.

## 7 Discussion

We mapped the succinctness, tractability and rapidity relations of several classical representations for quantum information. Common knowledge says that there is a trade-off between the succinctness of a representation, and the tractability of its operations. By contrast, we find that e.g. `ADD`, `SLDD_x`, and `MPS` are each successively more succinct and more rapid. *This demonstrates the superiority of the rapidity measure over the tractability measure*, because the latter seems to indicate instead that `SLDD_x` is less efficient than `ADD` for e.g. the addition operation. This is because `SLDD_x` is only intractable, when the representation is already exponentially smaller than `ADD`. Surprisingly, except for (Lai, Liu, and Yin 2017), rapidity is rarely used in knowledge compilation, which hitherto focussed mainly on succinctness and tractability (Bart et al. 2014; Pipatsrisawat and Darwiche 2008; Fargier and Marquis 2014; Fargier, Marquis, and Niveau 2013; Fargier and Marquis 2008a,b; Bova et al. 2016; Koriche et al. 2013; Onaka et al. 2025).

In practice, `SLDD_x` has achieved striking performance on realistic benchmarks (Zulehner and Wille 2018) due to a successful sustained effort to optimize the software implementation. But we emphasize that `MPS` was not developed with the intention for circuit simulation but for quantum simulation, and vice versa for `SLDD_x`. Therefore, empirical research is needed to compare the relative performance of these representations on the various application domains.

Aside from RBM, our investigation mainly focussed on so-called *ordered* representations that order qubits linearly (`MPS`, `ADD`, `SLDD_x`, `LIMDD`), just like (Fargier et al. 2014). Future extensions could include other representations that focus on higher-dimensional systems such as tree tensor networks (Orús 2014), PEPS & MERA (Verstraete and Cirac 2004), affine algebraic decision diagrams (Sanner and McAllester 2005), Context-Free-Language Ordered Binary Decision Diagrams (Sistla, Chaudhuri, and Reps 2023a,c,b) and Quantum Branching Programs (Ablayev, Gainutdinova, and Karpinski 2001) (despite being somewhat different, accepting a language rather than a quantum state). Like (Fargier et al. 2014), we focused on asymptotic behavior; in the future, we also aim to include numerical stability, which is known to influence diagram size in practice (Zulehner, Hillmich, and Wille 2019; Brand et al. 2025).

## Acknowledgements

This publication is part of the project Divide & Quantum (with project number 1389.20.241) of the research program NWA-ORC which is (partly) financed by the Dutch Research Council (NWO). This work was supported by the Dutch National Growth Fund, as part of the Quantum Delta NL program. The second author acknowledges the support received through the NWO Quantum Technology program (project number NGF.1582.22.035).

## References

- Aaronson, S.; and Gottesman, D. 2004. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5).
- Ablayev, F.; Gainutdinova, A.; and Karpinski, M. 2001. On computational power of quantum branching programs. In *Fundamentals of Computation Theory: FCT 2001, Proceedings 13*, 59–70. Springer.
- Ardeshir-Larijani, E.; Gay, S. J.; and Nagarajan, R. 2014. Verification of Concurrent Quantum Protocols by Equivalence Checking. In *TACAS*, 500–514. Springer.
- Audemard, G.; Koriche, F.; and Marquis, P. 2020. On tractable XAI queries based on compiled representations. In *Principles of Knowledge Representation and Reasoning*, volume 17, 838–849.
- Bahar, R. I.; Frohm, E. A.; Gaona, C. M.; Hachtel, G. D.; Macii, E.; Pardo, A.; and Somenzi, F. 1997. Algebraic decision diagrams and their applications. *Formal methods in system design*, 10(2-3): 171–206.
- Bart, A.; Koriche, F.; Lagniez, J.-M.; and Marquis, P. 2014. Symmetry-driven decision diagrams for knowledge compilation. In *ECAI 2014*, 51–56. IOS Press.
- Bärtschi, A.; and Eidenbenz, S. 2019. Deterministic preparation of Dicke states. In *Fundamentals of Computation Theory: 22nd International Symposium, FCT 2019, Proceedings 22*, 126–139. Springer.
- Benedetti, M.; Lloyd, E.; Sack, S.; and Fiorentini, M. 2019. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4): 043001.
- Bollig, B.; and Wegener, I. 1996. Improving the variable ordering of OBDDs is NP-complete. *IEEE Transactions on Computers*, 45: 993–1002.
- Bova, S. 2016. SDDs are exponentially more succinct than OBDDs. In *AAAI 2016*, volume 30.
- Bova, S.; Capelli, F.; Mengel, S.; and Slivovsky, F. 2016. Knowledge Compilation Meets Communication Complexity. In *IJCAI*, volume 16, 1008–1014.
- Brand, S.; Quist, A.-J.; van Dijk, R. M.; and Laarman, A. 2025. Numerical Errors in Quantitative System Analysis With Decision Diagrams. In *FORMATS*, 371–388. Springer.
- Bravyi, S.; and Kitaev, A. 2005. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Phys. Rev. A*, 71: 022316.
- Bryant, R. E. 1986. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Trans. Computers*, 35(8): 677–691.
- Burgholzer, L.; Kueng, R.; and Wille, R. 2021. Random stimuli generation for the verification of quantum circuits. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, 767–772.
- Burgholzer, L.; Ploier, A.; and Wille, R. 2022. Simulation paths for quantum circuit simulation with decision diagrams what to learn from tensor networks, and what not. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(4): 1113–1122.
- Burgholzer, L.; and Wille, R. 2020. Advanced equivalence checking for quantum circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(9): 1810–1824.
- Carleo, G.; and Troyer, M. 2017. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325): 602–606.
- Chen, J.; Cheng, S.; Xie, H.; Wang, L.; and Xiang, T. 2018. Equivalence of restricted Boltzmann machines and tensor network states. *Physical Review B*, 97(8): 085104.
- Clarke, E. M.; McMillan, K. L.; Zhao, X.; Fujita, M.; and Yang, J. 1993. Spectral transforms for large boolean functions with applications to technology mapping. In *Proceedings of the 30th international Design Automation Conference*, 54–60.
- Darwiche, A. 2001. Decomposable negation normal form. *Journal of the ACM (JACM)*, 48(4): 608–647.
- Darwiche, A. 2011. SDD: a new canonical representation of propositional knowledge bases. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two*, 819–826. AAAI Press.
- Darwiche, A.; and Marquis, P. 2002. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17: 229–264.
- Dicke, R. H. 1954. Coherence in spontaneous radiation processes. *Physical review*, 93(1): 99.
- Dumoulin, V.; Goodfellow, I.; Courville, A.; and Bengio, Y. 2014. On the Challenges of Physical Implementations of RBMs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1).
- Dunjko, V.; and Briegel, H. J. 2017. Machine learning & artificial intelligence in the quantum domain. arXiv:1709.02779.
- Fargier, H.; and Marquis, P. 2008a. Extending the knowledge compilation map: Closure principles. In *ECAI 2008*, 50–54. IOS Press.
- Fargier, H.; and Marquis, P. 2008b. Extending the knowledge compilation map: Krom, Horn, affine and beyond. In *23th AAAI Conference on Artificial Intelligence (AAAI 2008)*, volume 2, 442–447. AAAI Press.
- Fargier, H.; and Marquis, P. 2014. Disjunctive closures for knowledge compilation. *Artificial Intelligence*, 216: 129–162.
- Fargier, H.; Marquis, P.; and Niveau, A. 2013. Towards a knowledge compilation map for heterogeneous representation language. In *23rd International Joint Conference on*

- Artificial Intelligence (IJCAI 2013)*, 877–883. AAAI Press: International Joint Conferences on Artificial Intelligence.
- Fargier, H.; Marquis, P.; Niveau, A.; and Schmidt, N. 2014. A knowledge compilation map for ordered real-valued decision diagrams. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.
- Fargier, H.; Marquis, P.; and Schmidt, N. 2013. Semiring Labeled Decision Diagrams, Revisited: Canonicity and Spatial Efficiency Issues. In *IJCAI*, 884–890.
- Foulkes, W.; Mitas, L.; Needs, R.; and Rajagopal, G. 2001. Quantum Monte Carlo simulations of solids. *Reviews of Modern Physics*, 73(1): 33.
- Glasser, I.; Sweke, R.; Pancotti, N.; Eisert, J.; and Cirac, I. 2019. Expressive power of tensor-network factorizations for probabilistic modeling. *Advances in Neural Information Processing Systems*, 32.
- Groote, J. F.; and Zantema, H. 2003. Resolution and binary decision diagrams cannot simulate each other polynomially. *Discrete Applied Mathematics*, 130(2): 157–171.
- Hong, X.; Zhou, X.; Li, S.; Feng, Y.; and Ying, M. 2022. A Tensor Network Based Decision Diagram for Representation of Quantum Circuits. *ACM Trans. Des. Autom. Electron. Syst.*, 27(6).
- Jerrum, M.; and Meeks, K. 2017. The parameterised complexity of counting even and odd induced subgraphs. *Combinatorica*, 37(5): 965–990.
- Knuth, D. E. 2005. *The Art of Computer Programming. Volume 4, Fascicle 1*. Addison-Wesley.
- Koriche, F.; Lagniez, J.-M.; Marquis, P.; and Thomas, S. 2013. Knowledge Compilation for Model Counting: Affine Decision Trees. In *IJCAI*, 947–953.
- Lai, Y.; Liu, D.; and Yin, M. 2017. New canonical representations by augmenting OBDDs with conjunctive decomposition. *Journal of Artificial Intelligence Research*, 58: 453–521.
- Lai, Y.; Meel, K. S.; and Yap, R. H. 2022. CCDD: A Tractable Representation for Model Counting and Uniform Sampling. *arXiv preprint arXiv:2202.10025*.
- Miller, D. M.; and Thornton, M. A. 2006. QMDD: A decision diagram structure for reversible and quantum circuits. In *36th International Symposium on Multiple-Valued Logic (ISMVL'06)*, 30–30. IEEE.
- Nielsen, M. A.; and Chuang, I. L. 2000. Quantum information and quantum computation. *Cambridge: Cambridge University Press*, 2(8): 23.
- Onaka, R.; Nakamura, K.; Nishino, M.; and Yasuda, N. 2025. Tensor Decomposition Meets Knowledge Compilation: A Study Comparing Tensor Trains with OBDDs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 15109–15117.
- Orús, R. 2014. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349: 117–158.
- Pipatsrisawat, K.; and Darwiche, A. 2008. New Compilation Languages Based on Structured Decomposability. In *AAAI*, volume 8, 517–522.
- Sanner, S.; and McAllester, D. 2005. Affine Algebraic Decision Diagrams (AADDs) and Their Application to Structured Probabilistic Inference. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI'05*, 1384–1390. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Schollwöck, U. 2011. The density-matrix renormalization group in the age of matrix product states. *Annals of physics*, 326(1): 96–192.
- Sistla, M.; Chaudhuri, S.; and Reps, T. 2023a. CFLOBDDs: Context-Free-Language Ordered Binary Decision Diagrams. *arXiv:2211.06818*.
- Sistla, M.; Chaudhuri, S.; and Reps, T. 2023b. Symbolic Quantum Simulation with Quasimodo. In Enea, C.; and Lal, A., eds., *Computer Aided Verification*, 213–225. Cham: Springer Nature Switzerland. ISBN 978-3-031-37709-9.
- Sistla, M.; Chaudhuri, S.; and Reps, T. 2023c. Weighted Context-Free-Language Ordered Binary Decision Diagrams. *arXiv:2305.13610*.
- Somenzi, F. 1999. Binary decision diagrams. In Broy, M.; and Steinbrüggen, R., eds., *Computational system design*, 303–366. IOS.
- Thanos, D.; Coopmans, T.; Laarman, A.; and . 2023. Fast equivalence checking of quantum circuits of Clifford gates. *arXiv:2308.01206*.
- Van den Nest, M.; Dehaene, J.; and De Moor, B. 2004. Graphical description of the action of local Clifford transformations on graph states. *Physical Review A*, 69(2): 022316.
- Verstraete, F.; and Cirac, J. I. 2004. Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions. *arXiv:cond-mat/0407066*.
- Viamontes, G. F.; Markov, I. L.; Hayes, J. P.; a, c.; and f, g. 2003. Improving gate-level simulation of quantum circuits. *Quantum Information Processing*, 2(5): 347–380.
- Vidal, G. 2003. Efficient classical simulation of slightly entangled quantum computations. *Physical review letters*, 91(14): 147902.
- Vinkhuijzen, L.; Coopmans, T.; Elkouss, D.; Dunjko, V.; and Laarman, A. 2023. LIMDD: A Decision Diagram for Simulation of Quantum Computing Including Stabilizer States. *Quantum*, 7: 1108.
- Vinkhuijzen, L.; Coopmans, T.; and Laarman, A. 2024. A Knowledge Compilation Map for Quantum Information. *arXiv:2401.01322*.
- Wilson, N. 2005. Decision diagrams for the computation of semiring valuations. In *Proceedings of the 19th international joint conference on Artificial intelligence*, 331–336.
- Zulehner, A.; Hillmich, S.; and Wille, R. 2019. How to efficiently handle complex values? Implementing decision diagrams for quantum computing. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 1–7. IEEE.
- Zulehner, A.; and Wille, R. 2018. Advanced simulation of quantum computations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(5): 848–859.