
Quantum Tensor Networks, Stochastic Processes, and Weighted Automata

Sandesh Adhikary*

Computer Science and Engineering
University of Washington
adhikary@cs.washington.edu

Siddarth Srinivasan*

Computer Science and Engineering
University of Washington
sidsrini@cs.washington.edu

Jacob Miller

Mila & DIRO
Université de Montréal
jmjacobmiller@gmail.com

Guillaume Rabusseau

CCAI chair - Mila & DIRO
Université de Montréal
grabus@iro.umontreal.ca

Byron Boots

Computer Science and Engineering
University of Washington
bboots@cs.washington.edu

Abstract

Modeling joint probability distributions over sequences has been studied from many perspectives. The physics community developed *matrix product states*, a tensor-train decomposition for probabilistic modeling, motivated by the need to tractably model many-body systems. But similar models have also been studied in the stochastic processes and weighted automata literature, with little work on how these bodies of work relate to each other. We address this gap by showing how stationary or uniform versions of popular quantum tensor network models have equivalent representations in the stochastic processes and weighted automata literature, in the limit of infinitely long sequences. We demonstrate several equivalence results between models used in these three communities: (i) uniform variants of matrix product states, Born machines and locally purified states from the quantum tensor networks literature, (ii) predictive state representations, hidden Markov models, norm-observable operator models and hidden quantum Markov models from the stochastic process literature, and (iii) stochastic weighted automata, probabilistic automata and quadratic automata from the formal languages literature. Such connections may open the door for results and methods developed in one area to be applied in another.

1 Introduction

Matrix product states (MPS) were first developed by the physics community as compact representations of often intractable wave functions of complex quantum systems (Perez-Garcia et al., 2006; Klümper et al., 1993; Fannes et al., 1992), in parallel with the equivalent tensor-train decomposition (Oseledets, 2011) developed in applied mathematics for high-order tensors. These tensor network models have been gaining popularity in machine learning, especially as means of compressing highly-parameterized models (Novikov et al., 2015; Garipov et al., 2016; Yu et al., 2017; Novikov et al., 2014). There has also been recent interest in directly connecting ideas and methods from quantum tensor networks to machine learning (Stoudenmire and Schwab, 2016; Han et al., 2018; Guo et al., 2018; Huggins et al., 2019). In particular, tensor networks have been used for probabilistic modeling as parameterizations of joint probability tensors (Glasser et al., 2019; Miller et al., 2021; Stokes and Terilla, 2019). But the same problem has also been studied from various other perspectives. Notably, observable operator models (Jaeger, 2000) or predictive state representations (PSRs) (Singh et al., 2004) from the machine learning literature and stochastic weighted automata (Balle et al., 2014b) are approaches to tackle essentially the same problem. While Thon and Jaeger (2015) provide an overview discussing connections between PSRs and stochastic weighted automata, their connection to MPS has not been extensively explored. At the same time, there exist many variants of tensor network models related to MPS that can be used for probabilistic modeling. Glasser et al. (2019) recently provided a thorough investigation of the relative expressiveness of various tensor networks for the *non-uniform*

case (where cores in the tensor decomposition need not be identical). However, to the best of our knowledge, similar relationships have not yet been established for the *uniform* case. We address these issues by examining how various quantum tensor networks relate to aforementioned work in different fields, and we derive a collection of results analyzing the relationships in expressiveness between uniform MPS and their various subclasses.

The uniform case is important to examine for a number of reasons. The inherent weight sharing in uniform tensor networks leads to particularly compact models, especially when learning from highly structured data. This compactness becomes especially useful when we consider physical implementations of tensor network models in quantum circuits. For instance, Glasser et al. (2019) draw an equivalence between local quantum circuits and tensor networks; network parameters define gates that can be implemented on a quantum computer for probabilistic modeling. Uniform networks have fewer parameters, corresponding to a smaller set of quantum gates and greater ease of implementation on resource constrained near-term quantum computers. Despite the many useful properties of uniformity, the tensor-network literature tends to focus more on non-uniform models. We aim to fill this gap by developing expressiveness relationships for uniform variants.

We expect that the connections established in this paper will also open the door for results and methods in one area to be applied in another. For instance, one of the proof strategies we adopt is to develop expressiveness relationships between subclasses of PSRs, and show how they also carry over to equivalent uniform tensor networks. Such cross fertilization also takes place at the level of algorithms. For instance, the learning algorithm for locally purified states (LPS) employed in Glasser et al. (2019) does not preserve uniformity of the model across time-steps, or enforce normalization constraints on learned operators. With the equivalence between uniform LPS and hidden quantum Markov models (HQMMs) established in this paper, the HQMM learning algorithm from Adhikary et al. (2020), based on optimization over the Stiefel manifold, can be adapted to learn uniform LPS *while enforcing all appropriate constraints*. Similarly, spectral algorithms that have been developed for stochastic process models such as hidden Markov models (HMMs) and PSRs (Hsu et al., 2012; Siddiqi et al., 2010; Bailly et al., 2009) could also be adapted to learn uniform LPS and uniform MPS models. Spectral algorithms typically come with consistency guarantees, along with rigorous bounds on sample complexity. Such formal guarantees are less common in tensor network methods, such as variants of alternating least squares (Oseledets, 2011) or density matrix renormalization group methods (White, 1992). On the other hand, tensor network algorithms tend to be better suited for very high-dimensional data; presenting an opportunity to adapt

them to scale up algorithms for stochastic process models.

Finally, one of our key motivations is to simply provide a means of translating between similar models developed in different fields. While prior works (Glasser et al., 2019; Kliesch et al., 2014; Critch and Morton, 2014) have noted similarities between tensor networks, stochastic processes and weighted automata, many formal and explicit connections are still lacking, especially in the context of model expressiveness. It is still difficult for practitioners in one field to verify that the model classes they have been working with are indeed used elsewhere, given the differences in nomenclature and domain of application; simply having a thesaurus to rigorously translate between fields can be quite valuable. Such a thesaurus is particularly timely given the growing popularity of tensor networks in machine learning. We hope that the connections developed in this paper will help bring together complementary advances occurring in these various fields.

Summary of Contributions In Section 2, we demonstrate that uniform Matrix product states (uMPS) are equivalent to predictive state representations and stochastic weighted automata, when taken in the *non-terminating limit* (where we evaluate probabilities sufficiently away from the end of a sequence). Section 3 presents the known equivalence between uMPS with non-negative parameters, HMMs, and probabilistic automata, to show in Section 4 that another subclass of uMPS called Born machines (BM) (Han et al., 2018) is equivalent to norm-observable operator models (NOOM) (Zhao and Jaeger, 2010) and quadratic weighted automata (QWA) (Bailly, 2011). We also demonstrate that uBMs and NOOMs are relatively restrictive model classes in that there are HMMs with no equivalent finite-dimensional uBM or NOOM (HMMs $\not\subseteq$ NOOMs/uBMs). Finally, in Section 5, we analyze a broadly expressive subclass of uMPS known as locally purified states (LPS), demonstrate its equivalence to hidden quantum Markov models (HQMMs), and discuss the open question of how the expressiveness of uLPS relates to that of uMPS. We thus develop a unifying perspective on a wide range of models coming from different communities, providing a rigorous characterization of how they are related to one another, as illustrated in Figures 1 and 2. The proofs for all theorems are provided in the Appendix. In our presentation, we routinely point out connections between tensor networks and relevant concepts in physics. However, we note that these models are not restricted to this domain. Furthermore, we focus on models with a finite number of outcomes or observations, also called alphabets in the formal language community. This is the most commonly used setting for the models we consider, and does not really restrict the applicability of our results.

Notation We use bold-face for matrix and tensor operators (e.g. \mathbf{A}), arrows over symbols to denote vectors

(e.g. \vec{x}), and plain non-bold symbols for scalars. The vector-arrows are also used to indicate vectorization (column-first convention) of matrices. We frequently make use of the ones matrix $\mathbf{1}$ (filled with 1s) and the identity matrix \mathbf{I} , as well as their vectorizations $\vec{\mathbf{1}}$ and $\vec{\mathbf{I}}$. We use overhead bars to denote complex conjugates (e.g. $\bar{\mathbf{A}}$) and \dagger for the conjugate transpose ($\bar{\mathbf{A}}^T = \mathbf{A}^\dagger$). Finally, $\text{tr}(\cdot)$ denotes the trace operation applied to matrices, and \otimes denotes the Kronecker product.

2 Uniform Matrix Product States

Given a sequence of N observations, where each outcome can take d_i values, the joint probability of any particular sequence y_1, \dots, y_N can be written using the following tensor-train decomposition, which gives an MPS:

$$P(y_1, \dots, y_N) \propto \text{MPS}_{y_1, \dots, y_N} = \mathbf{A}^{[N], y_N} \mathbf{A}^{[N-1], y_{N-1}} \dots \mathbf{A}^{[2], y_2} \mathbf{A}^{[1], y_1} \quad (1)$$

where each $\mathbf{A}^{[i]}$ is a three-mode tensor *core* of the MPS containing d_i slices, with the matrix slice associated with outcome Y_i denoted by $\mathbf{A}^{[i], y_i}$. The above joint probability distribution must be normalized through a partition function. A common strategy to efficiently compute this partition function is to convert the tensor chain into a canonical form wherein expensive tensor contractions reduce to the identity. Each matrix slice $\mathbf{A}^{[i], y_i}$ is a $D_{i+1} \times D_i$ matrix, and the conventional choice (which we use in this paper) of *open boundary conditions* is to set $D_0 = D_N = 1$ (i.e. $\mathbf{A}^{[1], y_1}$ and $\mathbf{A}^{[N], y_N}$ are column and row vectors respectively). MPS with open boundaries are equivalent to tensor train (TT) decompositions, and we will define them over the complex field, a choice common in quantum physics and tensor network settings. The maximal value of $D = \max_k D_k$ is also called the bond-dimension or the TT-rank (Glasser et al., 2019) of the MPS. For fixed dynamics, this will lead the MPS cores $\mathbf{A}^{[i]}$ to be identical.

In this paper, we will focus on the ‘‘uniform’’ case of identical cores, i.e., a uniform MPS or uMPS. uMPS models were first developed in the quantum physics community (Perez-Garcia et al., 2006; Vanderstraeten et al., 2019), although employing a different probabilistic correspondence (Born machines as discussed later) than described below. As we will discuss, this corresponds naturally to Markovian dynamical systems; the notion of *past*

An alternate choice, *periodic boundary conditions*, sets $D_0 = D_N \geq 1$ and uses a trace operation to evaluate the product of matrices in Equation 1. MPS with periodic boundaries are equivalent to the tensor ring decomposition (Mickelin and Karaman, 2018).

Operational characterizations of the bond dimension have been developed in quantum physics, in terms of entanglement (Eisert et al., 2010) or the state space dimension of recurrent many-body dynamics which generate the associated wavefunction (Schoen et al., 2005).

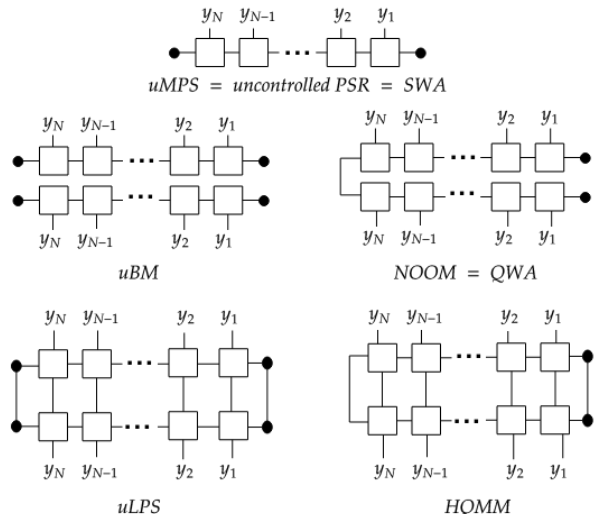


Figure 1: **Tensor Network Diagrams.** White squares correspond to tensors with as many modes as lines emanating from them, and black end dots indicate boundary vectors. A connecting line represent contraction along a mode, while adjacent tensors without connecting lines are multiplied together via the Kronecker product. Connecting lines (without black dots) at the boundaries represent the application of the identity.

being independent of future given the present is encoded by the tensor train structure where each core only has two neighbours. While an MPS is inherently defined with respect to a fixed sequence length, a uMPS can be applied to sequences of arbitrary fixed or infinite length (Cirac and Sierra, 2010). As there should be no distinction between the cores at different time steps in a uMPS, a natural representation is to fix two boundary vectors ($\vec{\sigma}$, $\vec{\rho}_0$), leading to the following decomposition of the joint probability:

$$P(y_1, \dots, y_N) = \text{uMPS}_{y_1, \dots, y_N} = \vec{\sigma}^\dagger \mathbf{A}^{y_N} \mathbf{A}^{y_{N-1}} \dots \mathbf{A}^{y_2} \mathbf{A}^{y_1} \vec{\rho}_0 \quad (2)$$

To explore connections with arbitrary-length PSRs and WFAs, we will particularly focus on the *non-terminating limit*. Consider that if we wished to compute the probability of some subsequence from $t = 1, \dots, T$ of the N -length uMPS ($T < N$), we could compute $P(y_1, \dots, y_T) = \sum_{y_N} \dots \sum_{y_{T+1}} P(y_1, \dots, y_T, y_{T+1}, \dots, y_N)$. The non-terminating limit is essentially when we consider the uMPS to be infinitely long, i.e., we compute the probability of the subsequence in the limit as $N \rightarrow \infty$.

Definition 1 (Non-terminating uMPS). A non-terminating uMPS is an infinitely long uMPS where we can compute the probability of any sequence $P(y_1, \dots, y_T)$ of length T by marginalizing over infinitely many future observations, i.e. $P(y_1, \dots, y_T) = \lim_{N \rightarrow \infty} \sum_{y_N} \dots \sum_{y_{T+1}} P(y_1, \dots, y_T, y_{T+1}, \dots, y_N)$.

This is a natural approach to modeling arbitrary length sequences with Markovian dynamics; intuitively, if given an identical set of tensor cores at each time step, the probability of a sequence should not depend on how far it is from the ‘end’ of the sequence. Similar notions are routinely used in machine learning and physics. In machine learning, it is common to discard the first few entries of sequences as “burn-in” to allow systems to reach their stationary distribution. In our case, the burn is being applied to the end of the sequence. The non-terminating limit is also similar to the “thermodynamic limit” employed in many-body physics, which marginalizes over an infinite number of future *and* past observations (Vanderstraeten et al., 2019). Such limits reflect the behavior seen in the interior of large systems, and avoid more complicated phenomena which arise near the beginning or end of sequences.

2.1 The Many Names of Matrix Product States

While connections between MPS and hidden Markov models (HMM) have been widely noted, we point out that non-terminating uMPS models have been studied from various perspectives, and are referred to by different names in the literature, such as stochastic weighted finite automata (stochastic WFA) (Balle et al., 2014a), quasi-realizations (Vidyasagar, 2011), observable operator models (OOM) (Jaeger, 2000), and (uncontrolled) predictive state representations (PSR) (Singh et al., 2004). The latter three models are exactly identical (we just refer to them as uncontrolled PSRs in this paper) and come from the stochastic processes perspective, while stochastic WFA are slightly different in their formulation, in that they are more similar to (terminating) uMPS (see below). Thon and Jaeger (2015) detail a general framework of *sequential systems* to study how PSRs and WFA relate to each other.

Predictive State Representations We write the stochastic process defined by an n -dimensional predictive state representation over a set of discrete observations \mathcal{O} as a tuple $(\mathbb{C}^n, \vec{\sigma}, \{\tau_y\}_{y \in \mathcal{O}}, \vec{x}_0)$. The initial state $\vec{x}_0 \in \mathbb{C}^n$ is normalized, as enforced by the linear evaluation functional $\vec{\sigma}$, i.e., $\vec{\sigma}^\dagger \vec{x}_0 = 1$, and the observable operators are constrained to have normalized marginals over observations $\vec{\sigma}^\dagger \sum_y \tau_y = \vec{\sigma}^\dagger$, i.e., $\vec{\sigma}^\dagger$ is a fixed point of the *transfer operator* $\sum_y \tau_y$. The probability of arbitrary length sequences $y_1, \dots, y_T \in \mathcal{O}^T$ is computed as $\vec{\sigma}^\dagger \tau_{y_T} \dots \tau_{y_1} \vec{x}_0$, which should be non-negative for any sequence. Note that we simply require this to hold for a valid PSR; we do not explicitly enforce constraints to ensure this. This joint probability computation is identical to Equation 2, where evaluation functional $\vec{\sigma}$ and the initial state \vec{x}_0 are analogous to the left and right boundary vectors of the uMPS, and the *observable operators* τ_y correspond to the matrix slices \mathbf{A}^y . In this sense, both uMPS and PSRs define tensor-train decompositions of joint distributions for a given fixed num-

ber of observations T . The only difference is that a uMPS does not require its evaluation functional to be the fixed point of its transfer operator. However, as we now discuss, any arbitrary uMPS evaluation functional will eventually converge to the fixed point of its transfer operator in the non-terminating limit. The fixed point then becomes the *effective* evaluation functional of the uMPS in this limit.

Since PSRs were formulated with dynamical systems in mind, we typically consider sequences of *arbitrary* length, whose probabilities are determined via a hidden state which evolves under a time-invariant update rule: the state update conditioned on an observation y_t is computed as $\vec{x}_t = \frac{\tau_{y_t} \vec{x}_{t-1}}{\vec{\sigma}^\dagger \tau_{y_t} \vec{x}_{t-1}}$ and the probability of an observation y_t is $P(y_t | \vec{x}_t) = \vec{\sigma}^\dagger \tau_{y_t} \vec{x}_t$. This allows us to deal more flexibly with arbitrary length sequences as compared to a generic uMPS. This flexibility for arbitrary-length sequences is precisely why we consider non-terminating uMPS: we can compute the conditional probability of a sequence $P(y_t | y_{1:t-1})$ by marginalizing over all possible future observations with a relatively simple equation:

$$\begin{aligned} P(y_t | y_{1:t-1}) &= \frac{\sum_{y_N, \dots, y_{t+1}} P(y_N, \dots, y_{t+1}, y_t, y_{t-1}, \dots, y_1)}{\sum_{y_N, \dots, y_t} P(y_N, \dots, y_t, y_{t-1}, \dots, y_1)} \\ &= \frac{\vec{\sigma}^\dagger \left(\sum_y \tau_y \right)^{N-(t+1)} \tau_{y_t} \dots \tau_1 \vec{\rho}_0}{\vec{\sigma}^\dagger \left(\sum_y \tau_y \right)^{N-t} \tau_{y_{t-1}} \dots \tau_1 \vec{\rho}_0} \end{aligned} \quad (3)$$

Thus the *effective* evaluation functional $\vec{\sigma}^\dagger \left(\sum_y \tau_y \right)^{N-t}$ is a function of time and so different at every time step in general. However, if the transfer operator $\tau = \sum_y \tau_y$ has some fixed point $\vec{\sigma}_*$, i.e., $\vec{\sigma}_*^\dagger \tau = \vec{\sigma}_*^\dagger$, then the effective evaluation functional at timestep t (which is $N-t$ steps away from the left boundary of the uMPS) will eventually converge to $\vec{\sigma}_*$ given a long enough sequence. So, as long as we remain sufficiently far from the end of a sequence, the particular choice of the the left boundary vector does not matter.

Given that the non-terminating limit effectively replaces the uMPS evaluation functional $\vec{\sigma}$ by the fixed point $\vec{\sigma}_*$, consider what happens if we require $\vec{\sigma} = \vec{\sigma}_*$ to begin with, as is the case for PSRs. In this case, our effective evaluation functional remains independent of t , permitting a simple recursive state update rule that does not require fixing a prior sequence length or marginalizing

In fact, the effective evaluation functional will converge at an exponential rate towards the fixed point, so that $\|\vec{\sigma}_t - \vec{\sigma}_*\|^2 = \mathcal{O}(\exp \frac{N-t}{\xi})$, with a “correlation length” $\xi \simeq (1 - |\lambda_2|/|\lambda_1|)^{-1}$ set by the ratio of the largest and second largest eigenvalues of the transfer operator (Orús, 2014). Transfer operators with non-degenerate spectra can always be rescaled to have a unique fixed point, while those with degenerate spectra form a measure zero subset.

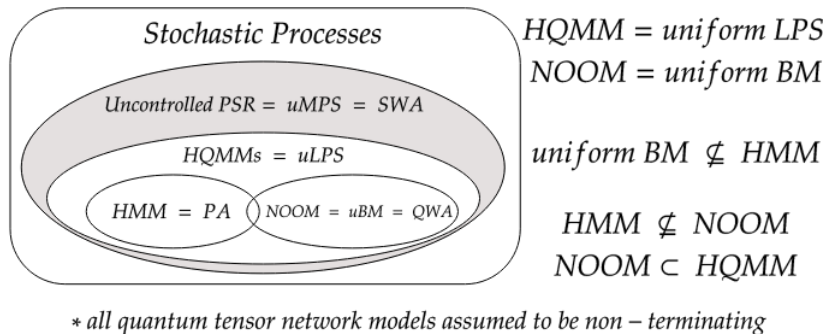


Figure 2: **Expressiveness Relationships Between Models:** Subset relationships between stochastic process models, non-terminating uniform quantum tensor networks, and weighted automata, along with a summary of new relationships established in this paper. The grey area is potentially empty.

over future observations. In this sense, a non-terminating uMPS is strictly equivalent to a PSR, a relationship which we will see holds between several other model families.

Theorem 2.1. *Non-terminating uniform matrix product states are equivalent to uncontrolled predictive state representations.*

If we do not consider the non-terminating limit of a uMPS, the subsequence length and boundary choice will affect the probability computed. Then, we technically only have an equivalence with PSRs for a fixed sequence length (when the evaluation functional is a fixed point of the transfer operator) and no notion of recursive state update.

Stochastic Weighted Automata A weighted automaton (WA) is a tuple $(\mathbb{C}^n, \vec{\sigma}, \{\tau_y\}_{y \in \mathcal{O}}, \vec{x}_0)$ which computes a function $f(y_1, \dots, y_T) = \vec{\sigma}^\dagger \tau_{y_T} \dots \tau_{y_1} \vec{x}_0$. In contrast with PSRs, no constraints are enforced on the weights of a weighted automaton in general. A weighted automaton is *stochastic* if it computes a probability distribution. These models constitute another class of models equivalent to PSRs and uMPS, and represent probability distributions over sequences of symbols from an alphabet \mathcal{O} . As discussed earlier, we focus on models with a finite number of alphabets. It is worth mentioning that the semantics of the probabilities computed by PSRs and stochastic WAs can differ: while PSR typically maintain a recursive state and are used to compute the probability of a given sequence conditioned on some past sequence, stochastic WA are often used to compute the joint distributions over the set of all possible finite length sequences (just as in uMPS). We refer the reader to Thon and Jaeger (2015) for a unifying perspective.

3 Non-Negative uMPS, Hidden Markov Models, and Probabilistic Automata

We first point out a well-known connection between hidden Markov models (HMM) and matrix product

states (Kliesch et al., 2014; Critch and Morton, 2014). We refer to uMPS where all tensor cores and boundary vectors are non-negative as *non-negative uMPS*. HMMs have been extensively studied in machine learning and are a common approach to modeling discrete-observation sequences where an unobserved hidden state undergoes Markovian evolution (where the future is independent of past given present) and emits observations at each time-step (Rabiner and Juang, 1986). HMMs can be thought of as a special case of PSRs where all parameters are constrained to be non-negative. Such models are usually characterized by an initial *belief state* \vec{x}_0 and column-stochastic transition and emission matrices \mathbf{A} and \mathbf{C} . Formally, we give the following definition:

Definition 2 (Hidden Markov Model). An n -dimensional hidden Markov model for a set of discrete observations \mathcal{O} is a stochastic process described by the tuple $(\mathbb{R}^n, \mathbf{A}, \mathbf{C}, \vec{x}_0)$. The transition matrix $\mathbf{A} \in \mathbb{R}_{\geq 0}^{n \times n}$ and the emission matrix $\mathbf{C} \in \mathbb{R}_{\geq 0}^{|\mathcal{O}| \times n}$ are non-negative and column stochastic i.e. $\vec{\mathbf{1}}^T \mathbf{A} = \vec{\mathbf{1}}^T \mathbf{C} = \vec{\mathbf{1}}^T$. The initial state $\vec{x}_0 \in \mathbb{R}_{\geq 0}^n$ is also non-negative and is normalized $\|\vec{x}_0\|_1 = \vec{\mathbf{1}}^T \vec{x}_0 = 1$.

The state transitions through the simple linear update $\vec{x}'_t = \mathbf{A} \vec{x}_{t-1}$. To condition on observation y , we construct the diagonal matrix $\text{diag}(\mathbf{C}_{(y,\cdot)})$ from the y^{th} row of \mathbf{C} , and perform a normalized update $\vec{x}_t | y_t = \frac{\text{diag}(\mathbf{C}_{(y,\cdot)}) \vec{x}'_t}{\vec{\mathbf{1}}^T \text{diag}(\mathbf{C}_{(y,\cdot)}) \vec{x}'_t}$. This multi-step filtering process can be simplified using an alternative representation with *observable operators* as $\mathbf{T}_y = \text{diag}(\mathbf{C}_{(y,\cdot)}) \mathbf{A}$, where we rewrite the normalization constraints on operators as $\mathbf{1}^T \sum_y \mathbf{T}_y = \mathbf{1}^T$. Then, we recover a recursive state update $\vec{x}_t = \frac{\mathbf{T}_{y_t-1} \vec{x}_{t-1}}{\vec{\mathbf{1}}^T \mathbf{T}_{y_t-1} \vec{x}_{t-1}}$. Clearly, HMMs are a special case of PSRs where the parameters are restricted to be non-negative, and a special case of uMPS when the left boundary $\vec{\sigma}^\dagger = \mathbf{1}^T$, the right boundary $\vec{\rho}_0 = \vec{x}_0$, and the tensor core slice $A^y = \mathbf{T}_y$.

Probabilistic Automata Lastly, non-negative uMPS are equivalent to probabilistic automata from formal language theory (Denis and Esposito, 2008, Section 4.2), which are in essence weighted automata where transition matrices need to satisfy stochasticity constraints. The strict equivalence between probabilistic automata and HMMs is proved in Dupont et al. (2005, Proposition 8) (see also Section 2.2 in Balle et al. (2014b)). In addition, it is known that non-negative uMPS are strictly less expressive than general uMPS for representing probability distributions; a proof of this result in the context of formal languages can be found in Denis and Esposito (2008). We give a brief discussion of this next.

3.1 The Negative Probability Problem and the Expressiveness of Finite PSRs

As noted by several authors (Jaeger, 2000; Adhikary et al., 2020), PSRs lack a constructive definition; the definition of a PSR simply demands that the probabilities produced by the model be non-negative without describing constraints that can achieve this. Indeed, this is the cost of relaxing the non-negativity constraint on HMMs; it is undecidable whether a given set of PSR parameters will assign a negative probability to some arbitrary-length sequence (Denis and Esposito, 2004; Wiewiora, 2008), an issue known as the *negative probability problem* (NPP). A similar issue arises in the many-body physics setting, where the analogous question of whether a general matrix product operator describes a non-negative quantum density operator is also undecidable (Kliesch et al., 2014). In the special case where all PSR parameters are non-negative, we have a sufficient condition for generating valid probabilities, namely that the PSR is a hidden Markov model. Otherwise, the best available approach to characterize valid states for PSRs is whether they define a pointed convex cone (that includes the initial state) which is closed under its operators, and all points in it generate valid probabilities (Heller, 1965; Jaeger, 2000; Adhikary et al., 2020).

While this undecidability is an inconvenient feature of PSRs, it turns out that constraining PSRs to have only non-negative entries comes with a reduction in expressive power; there are finite (bond/state) dimensional uMPS/PSRs which have no equivalent finite-dimensional HMM representations for arbitrary length sequences (for example, the probability clock in Jaeger (2000)). The general question of which uncontrolled PSRs have equivalent finite-dimensional HMMs (though not always discussed in those terms) is referred to by some as the *positive realization problem* (Benvenuti and Farina, 2004; Vidyasagar, 2011). A common approach is to use the result that a PSR has an equivalent finite-dimensional HMM if and only if the aforementioned convex cone of valid initial states $\{\vec{x}_0\}$ for a set of given operators $\vec{\sigma}^\dagger, \{\tau_y\}$ is k -polyhedral for some finite k (Jaeger, 2000).

There has been some work trying to investigate whether it is possible to maintain the superior expressiveness of uMPS/PSRs while avoiding the undecidability issue. Zhao and Jaeger (2010); Bailly (2011); Adhikary et al. (2020) explore this question in the machine learning context, while Glasser et al. (2019) consider this problem from the quantum tensor network perspective. We will explore these proposals shortly. When discussing the relative expressiveness of a model compared to a uMPS/PSR, if its bond dimension (i.e. state dimension) grows with sequence length, we say there is no equivalent parameterization of the uMPS/PSR distribution in this model class. In other words, we consider two model classes equivalent if any joint distribution over sequences that can be represented by a model in one class (with finite bond/latent dimension) can be represented exactly by a model in the other class (also with finite bond/latent dimension).

4 Uniform Born Machines, Norm-Observable Operator Models, and Quadratic Weighted Automata

Born machines (BMs) (Han et al., 2018) are a popular class of quantum tensor networks that model probability densities as the absolute-square of the outputs of a tensor-train decomposition, and hence always output valid probabilities. As with uMPS, we will work with uniform Born machines (uBMs) (Miller et al., 2021), for which the joint probability of N discrete random variables $\{Y_i\}_{i=1}^N$ is computed as follows (with boundary vectors $\vec{\alpha}$ and $\vec{\omega}_0$ sandwiching a sequence of identical cores \mathbf{A}):

$$P(y_1, \dots, y_N) = \text{uBM}_{y_1, \dots, y_N} = |\vec{\alpha}^\dagger \mathbf{A}^{y_N} \dots \mathbf{A}^{y_1} \vec{\omega}_0|^2 \quad (4)$$

We can re-write this decomposition showing uBMs to be special kinds of uMPS/PSR:

$$\begin{aligned} \text{uBM}_{y_1, \dots, y_N} &= \left| \vec{\alpha}^\dagger \mathbf{A}^{y_N} \dots \mathbf{A}^{y_1} \vec{\omega}_0 \right|^2 \\ &= \vec{\alpha}^\dagger \mathbf{A}^{y_N} \dots \mathbf{A}^{y_1} \vec{\omega}_0 \vec{\omega}_0^\dagger (\mathbf{A}^{y_1})^\dagger \dots (\mathbf{A}^{y_N})^\dagger \vec{\alpha} \quad (5) \\ &= \vec{\sigma}^\dagger \tau_{y_N} \dots \tau_{y_1} \vec{\rho}_0 \end{aligned}$$

where $\tau_y = \overline{\mathbf{A}^y} \otimes \mathbf{A}^y$, $\vec{\rho}_0 = \overline{\vec{\omega}_0} \otimes \vec{\omega}_0$, and $\vec{\sigma} = \overline{\vec{\alpha}} \otimes \vec{\alpha}$. This makes it clear that uBMs are a special class of MPS/PSRs, where the observable operators τ_y and boundary conditions must satisfy the additional requirement of having unit Kraus-rank (i.e., a symmetric unit Schmidt rank decomposition).

Norm Observable Operator Models Motivated by the NPP for PSRs, Zhao and Jaeger (2010) introduce norm-observable operator models or NOOMs. Coming from the PSR literature, they were designed to model joint

An operator \mathbf{A} has unit Kraus-rank if it has a decomposition $\mathbf{A} = \overline{\mathbf{X}} \otimes \mathbf{X}$.

distributions of observations as well as recursive state-updates to obtain conditional probabilities (analogous to PSRs in Section 2.1). They bear a striking resemblance to uniform Born machines (uBMs) and the connection has not been previously explored. Both NOOMs and uBMs associate probabilities with quadratic functions of the state vector, with NOOMs directly using the squared 2-norm of the state to determine observation probabilities. While NOOMs were originally defined on the reals, we use a more general definition over complex numbers.

Definition 3 (Norm-observable operator model). An n -dimensional norm-observable operator model for a set of discrete observations \mathcal{O} is a stochastic process described by the tuple $(\mathbb{C}^n, \{\phi_y\}_{y \in \mathcal{O}}, \vec{\psi}_0)$. The initial state $\vec{\psi}_0 \in \mathbb{C}^n$ is normalized by having unit 2-norm i.e. $\|\vec{\psi}_0\|_2^2 = 1$. The operators $\phi_y \in \mathbb{C}^{n \times n}$ satisfy $\sum_y \phi_y^\dagger \phi_y = \mathbb{I}$.

These models avoid the NPP by using the 2-norm of the state to recover probability which, unlike for HMMs, is insensitive to the use of negative parameters in the matrices ϕ_y . We write the joint probability of a sequence as computed by a NOOM and manipulate it using using the relationship between 2-norm and trace to show:

$$\begin{aligned} P(y_1, \dots, y_N) &= \text{NOOM}_{y_1, \dots, y_N} \\ &= \left\| \phi_{y_N} \cdots \phi_{y_1} \vec{\psi}_0 \right\|_2^2 \\ &= \text{tr}(\phi_{y_N} \cdots \phi_{y_1} \vec{\psi}_0 \vec{\psi}_0^\dagger (\phi_{y_1})^\dagger \cdots (\phi_{y_N})^\dagger) \\ &= \vec{\mathbb{I}}^\dagger \tau_{y_N} \cdots \tau_{y_1} \vec{\rho}_0 \end{aligned} \quad (6)$$

where $\tau_y = \overline{\phi_y} \otimes \phi_y \in \mathbb{C}^{n^2 \times n^2}$ and $\vec{\rho}_0 = \overline{\vec{\psi}_0} \otimes \vec{\psi}_0$. Equation 6 shows that NOOMs are a special subset of PSRs/MPS, as every finite-dimensional NOOM has an equivalent finite-dimensional PSR $(\mathbb{C}^{n^2}, \vec{\mathbb{I}}, \{\overline{\phi_y} \otimes \phi_y\}_{y \in \mathcal{O}}, \overline{\vec{\psi}_0} \otimes \vec{\psi}_0)$ (Zhao and Jaeger, 2010). From a quantum mechanical perspective, the unit rank constraint on NOOM initial states can be framed as requiring the initial state to be a pure density matrix. We can also recursively update the state conditioned on observation y_t as $\vec{\psi}_t = \frac{\phi_{y_t} \vec{\psi}_{t-1}}{\|\phi_{y_t} \vec{\psi}_{t-1}\|_2}$, where y_t is observed with probability $P(y_t | \vec{\psi}_t) = \|\phi_{y_t} \vec{\psi}_{t-1}\|_2^2$.

Non-terminating uniform BMs are NOOMs
Note that the NOOM joint distribution in Equation 6 is almost identical to that of uBMs in Equation 5, with τ_y and $\vec{\rho}_0$ having unit Kraus rank; but the left boundary / evaluation functional $\vec{\sigma} = \vec{\alpha} \otimes \vec{\alpha}$ is replaced by $\vec{\sigma} = \vec{\mathbb{I}}$ and necessarily is full Kraus rank. So how can we reconcile these nearly identical models? Similar to our approach in Section 2.1, we can consider the uBM for an infinitely long sequence where the exact specification of the left boundary / evaluation functional ceases to matter in the non-terminating limit; the effective evaluation functional converges to the fixed point of the transfer operator and we have a notion of recursive state update. Assuming that

the uBM transfer operator is a similarity transform away from a trace-preserving quantum channel (i.e., which is normalized by satisfying $\sum_y \phi_y^\dagger \phi_y = \mathbb{I}$; see appendix for more details), we have that an arbitrary evaluation functional (with unit Kraus-rank) of such a uBM will eventually converge to $\vec{\mathbb{I}}$, the NOOM's evaluation functional:

Theorem 4.1. *Non-terminating uniform Born machines are equivalent to norm observable operator models.*

With the above equivalence, we now turn to the question of how the expressiveness of uBM/NOOMs compares to non-negative uMPS/HMMs. As we have seen, they are all special classes of uMPS, but with different constructions. Glasser et al. (2019) studied the expressiveness of *non-uniform* BMs, showing that there are finite-dimensional non-uniform BMs that cannot be modeled by finite dimensional non-uniform HMMs, and conjecture that the reverse direction is also true. Zhao and Jaeger (2010) showed by example the existence of a NOOM (and so a non-terminating uBM) that cannot be modeled by any finite-dimensional HMM. However, they left open the question of whether HMMs were a subclass of NOOMs. We answer this question in the following theorem, which also implies the latter corollary through its equivalence with non-terminating uBM.

Theorem 4.2 (HMM $\not\subseteq$ NOOM). *There exist finite-dimensional hidden Markov models that have no equivalent finite-dimensional norm-observable operator model.*

Corollary 4.1 (uBM $\not\subseteq$ HMM and HMM $\not\subseteq$ uBM). *There exist finite-dimensional non-terminating uniform Born machines that have no equivalent finite-dimensional hidden Markov models, and vice-versa.*

The proof (see appendix) relies on the fact that both NOOMs and HMMs are special cases of PSRs, and any two equivalent PSRs are related by a similarity transform. We show that there is no similarity transform between HMMs and NOOMs that preserves the normalization requirement of NOOM states.

Quadratic Weighted Automata Finally, we note that quadratic weighted automata (QWA) (Bailly, 2011), developed in the stochastic weighted automata literature, are equivalent to uBM. Bailly (2011) suggest that QWA $\not\subseteq$ HMM and that HMM $\not\subseteq$ QWA, but do not provide a proof. To the best of our knowledge, the proof we provide is the first to formally show the non-equivalence of QWA and HMM.

5 Locally Purified States and Hidden Quantum Markov Models

While uBMs/NOOMs are constructive models guaranteed to return valid probabilities, they still aren't expressive enough to capture all HMMs, a fairly general class.

Hence, it may be desirable to identify a construction that is more expressive than these models but still gives valid probabilities. Locally purified states (LPS) were proposed as a tensor-network model of discrete multivariate probability distributions inspired from techniques used in the simulation of quantum systems. Glasser et al. (2019) show that these models are not only strictly more expressive than non-uniform HMMs, but also correspond directly to local quantum circuits with ancillary qubits – serving as a guide to design quantum circuits for probabilistic modeling. We arrive at the LPS model from the MPS model essentially by marginalizing over an additional mode – called the “purification dimension” – in each of the MPS tensors. The rank of an LPS, also called its puri-rank, is defined the same way as the bond dimension (or TT-rank) for the MPS. The corresponding uniform LPS defines the unnormalized probability mass function over N discrete random variables $\{Y_i\}_{i=1}^N$ as follows:

$$\begin{aligned} P(y_1, \dots, y_N) &= \text{uLPS}_{y_1, \dots, y_N} \\ &= \left(\sum_{\beta_L} \bar{\mathbf{K}}_{\beta_L, L}^T \otimes \mathbf{K}_{\beta_L, L}^T \right) \left(\sum_{\beta} \bar{\mathbf{K}}_{\beta, y_N} \otimes \mathbf{K}_{\beta, y_N} \right) \cdots \\ &\cdots \left(\sum_{\beta} \bar{\mathbf{K}}_{\beta, y_1} \otimes \mathbf{K}_{\beta, y_1} \right) \left(\sum_{\beta_R} \bar{\mathbf{K}}_{\beta_R, R} \otimes \mathbf{K}_{\beta_R, R} \right) \end{aligned} \quad (7)$$

Hidden Quantum Markov Models Hidden quantum Markov models (HQMMs) were developed by Monras et al. (2010) as a quantum generalization of hidden Markov models that can model joint probabilities of sequences and also allow for recursive state updates we have described previously. Srinivasan et al. (2018b,a) specifically develop HQMMs by constructing quantum analogues of classical operations on graphical models, and show that HQMMs are a more general model class compared to HMMs. Adhikary et al. (2020) on the other hand develop HQMMs by relaxing the unit Kraus-rank constraint on NOOM operators and initial state. We give a formal definition of these models here (noting that the Choi matrix \mathbf{C}_y is a particular reshuffling of the sum of superoperators \mathbf{L}_y defined below (see Adhikary et al. (2020)):

Definition 4 (Hidden Quantum Markov Models). An n^2 -dimensional hidden quantum Markov model for a set of discrete observations \mathcal{O} is a stochastic process described by the tuple $(\mathbb{C}^{n^2}, \bar{\mathbb{I}}, \{\mathbf{L}_y\}_{y \in \mathcal{O}}, \vec{\rho}_0)$. The initial state $\vec{\rho}_0 \in \mathbb{C}^{n^2}$ is a vectorized unit-trace Hermitian PSD matrix of arbitrary rank, so $\bar{\mathbb{I}}^T \vec{\rho}_0 = 1$. The Liouville operators $\mathbf{L}_y \in \mathbb{C}^{n^2 \times n^2}$ (with corresponding Choi matrices \mathbf{C}_y) are trace-preserving (TP) i.e. $\bar{\mathbb{I}}^T (\sum_y \mathbf{L}_y) = \bar{\mathbb{I}}^T$, and completely positive (CP) i.e. $\mathbf{C}_y \succeq 0$.

The CP-TP condition on the operators \mathbf{L}_y implies that we can equivalently write it via the Kraus decomposition

as $\mathbf{L}_y = \left(\sum_{\beta} \mathbf{K}_{\beta, y} \otimes \bar{\mathbf{K}}_{\beta, y} \right)$, using Kraus operators $\mathbf{K}_{\beta, y}$ (Kraus, 1971; Adhikary et al., 2020). Intuitively, what makes HQMMs more general than NOOMs is that its state can be a vectorized density matrix of arbitrary rank and the superoperators can have arbitrary Kraus-rank, while NOOMs require both these ranks to be 1. With this in mind, we can write and manipulate the joint probability of a sequence of N observations as:

$$\begin{aligned} P(y_1, \dots, y_N) &= \text{HQMM}_{y_1, \dots, y_N} = \bar{\mathbb{I}}^T \mathbf{L}_{y_1} \cdots \mathbf{L}_{y_N} \vec{\rho}_0 \\ &= \bar{\mathbb{I}}^T \left(\sum_{\beta} \bar{\mathbf{K}}_{\beta, y_N} \otimes \mathbf{K}_{\beta, y_N} \right) \cdots \left(\sum_{\beta} \bar{\mathbf{K}}_{\beta, y_1} \otimes \mathbf{K}_{\beta, y_1} \right) \vec{\rho}_0 \end{aligned} \quad (8)$$

The joint probability computation makes it clear that HQMMs are a class of PSRs, and the manipulation shows how they are equivalent to a uLPS where the left boundary condition is $\bar{\mathbb{I}}$. We also compute the recursive state update conditioned on observation y as $\vec{\rho}_{t+1} = \frac{\mathbf{L}_y \vec{\rho}_t}{\bar{\mathbb{I}}^T \mathbf{L}_y \vec{\rho}_t}$ and the probability of an observation y is $P(y|\rho_t) = \bar{\mathbb{I}}^T \mathbf{L}_y \vec{\rho}_t$.

Non-terminating Uniform LPS are HQMMs Equation 8 shows that every HQMM is a uLPS, but we also consider in what sense every uLPS is an HQMM: the transfer operator of arbitrary CP maps with unit spectral radius is a similarity transform away from that of a CP-TP map (Perez-Garcia et al., 2006), so $\bar{\mathbb{I}}$ is related to such a fixed point by such a similarity transform. Thus, every non-terminating uLPS has an equivalent HQMM and allows for an HQMM-style recursive state update. This is the same reasoning behind the equivalence between non-terminating uBMs (with CP maps) and NOOMs (with CP-TP maps).

Theorem 5.1. *Non-terminating uniform locally purified states are equivalent to hidden quantum Markov models.*

While it is already known that HMMs are a strict subset of HQMMs (since HQMMs also contain NOOMs which cannot always be modeled by a HMM), Adhikary et al. (2020) left open the possibility that every HQMM could have an equivalent NOOM in some higher dimensional state space. In light of Theorem 4.2, we can say this is not possible as NOOMs do not capture HMMs, while HQMMs can.

Corollary 5.1 (NOOM \subset HQMM). *Finite dimensional norm-observable operator models are a strict subset of finite dimensional hidden quantum Markov models.*

We are not aware of any proposals from the weighted automata literature that are analogous to these uLPS/HQMMs.

This condition is necessary for probability distributions such as Equations 5 and 7 to be properly normalized.

Expressiveness of HQMMs (uLPS) and PSRs (uMPS) We have determined that HQMMs are the most constructive known subclass of PSRs (containing both NOOMs and HMMs), yet the question of whether there is a ‘gap’ between HQMMs and PSRs, i.e., if there is a PSR which has no finite-dimensional HQMM representation, is still open to the best of our knowledge. The results in Glasser et al. (2019) and De las Cuevas et al. (2013) show that MPS are more expressive than LPS in the *non-uniform* case, but their technique cannot be easily adapted to the uniform case. We are not aware of an example of a PSR with no equivalent finite-dimensional HQMM. A longer discussion of this problem is presented in Appendix 2.

6 Conclusion

We presented uniform matrix product states and their various subclasses, and showed how they relate to previous work in the stochastic processes and weighted automata literature. In discussing the relative expressiveness of various models, we discuss if we can find an *equivalent finite-dimensional parameterization* in another model class, but we do not discuss the relative compactness of various parameterizations. Glasser et al. (2019) do discuss this for the non-uniform case, and this could be an interesting direction to explore for the uniform case. We also speculate that the connections laid out here may make spectral learning algorithms commonly used for PSRs and weighted automata (Hsu et al., 2012; Balle et al., 2014a; Hefny et al., 2015) suitable for learning uMPS, and an algorithm for optimization on the Stiefel manifold (Adhikary et al., 2020) suitable for learning uLPS *with appropriate constraints*. Future work will involve adapting these algorithms so they can be transferred between the two fields.

We can extend our analyses to *controlled* stochastic processes. Controlled generalizations of uMPS may be developed through matrix product operators (Murg et al., 2008; Chan et al., 2016) that append an additional open index at each core of a uMPS, which we can associate with actions. We can also develop input-output versions of uniform tensor networks and uncontrolled stochastic process models, similar to input-output OOMs from Jaeger (1998). We briefly describe such extensions for HQMMs and uLPSs in Appendix 3, showing that they generalize recently proposed quantum versions of partially observable Markov decisions processes (Barry et al., 2014; Ying and Ying, 2014; Cidre, 2016). With this connection, we also find that the undecidability of perfect planning (determining if there exists a policy that can deterministically reach a goal state from an arbitrary initial state in finite steps) established for quantum POMDPs by Barry et al. (2014) extends to these generalizations as well. We leave a longer discussion for future work.

References

- Adhikary, S., Srinivasan, S., Gordon, G., and Boots, B. (2020). Expressiveness and learning of hidden quantum markov models. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*.
- Bailly, R. (2011). Quadratic weighted automata: Spectral algorithm and likelihood maximization. In *Asian Conference on Machine Learning*, pages 147–163.
- Bailly, R., Denis, F., and Ralaivola, L. (2009). Grammatical inference as a principal component analysis problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 33–40.
- Balle, B., Carreras, X., Luque, F. M., and Quattoni, A. (2014a). Spectral learning of weighted automata. *Machine learning*, 96(1-2):33–63.
- Balle, B., Hamilton, W., and Pineau, J. (2014b). Methods of moments for learning stochastic languages: Unified presentation and empirical comparison. In *International Conference on Machine Learning*, pages 1386–1394.
- Barry, J., Barry, D. T., and Aaronson, S. (2014). Quantum partially observable markov decision processes. *Phys. Rev. A*, 90:032311.
- Benvenuti, L. and Farina, L. (2004). A tutorial on the positive realization problem. *IEEE Transactions on automatic control*, 49(5):651–664.
- Chan, G., Keselman, A., Nakatani, N., Li, Z., and White, S. (2016). Matrix product operators, matrix product states, and ab initio density matrix renormalization group algorithms. *The Journal of chemical physics*, 145 1:014102.
- Cidre, G. A. (2016). Planning in a quantum system. *Masters Thesis, Carnegie Mellon University*.
- Cirac, J. and Sierra, G. (2010). Infinite matrix product states, conformal field theory, and the haldane-shastry model. *Physical Review B*, 81:104431.
- Critch, A. and Morton, J. (2014). Algebraic geometry of matrix product states. *Symmetry Integrability and Geometry-methods and Applications*, 10:095.
- De las Cuevas, G., Schuch, N., Pérez-García, D., and Cirac, J. I. (2013). Purifications of multipartite states: limitations and constructive methods. *New Journal of Physics*, 15(12):123021.
- Denis, F. and Esposito, Y. (2004). Learning classes of probabilistic automata. In *International Conference on Computational Learning Theory*, pages 124–139. Springer.
- Denis, F. and Esposito, Y. (2008). On rational stochastic languages. *Fundamenta Informaticae*, 86(1, 2):41–77.
- Dupont, P., Denis, F., and Esposito, Y. (2005). Links between probabilistic automata and hidden markov models: probability distributions, learning models and induction algorithms. *Pattern recognition*, 38(9):1349–1371.

- Eisert, J., Cramer, M., and Plenio, M. B. (2010). Colloquium: Area laws for the entanglement entropy. *Reviews of Modern Physics*, 82(1):277.
- Fannes, M., Nachtergaele, B., and Werner, R. F. (1992). Finitely correlated states on quantum spin chains. *Communications in mathematical physics*, 144(3):443–490.
- Garipov, T., Podoprikin, D., Novikov, A., and Vetrov, D. (2016). Ultimate tensorization: compressing convolutional and fc layers alike. *arXiv preprint arXiv:1611.03214*.
- Glasser, I., Sweke, R., Pancotti, N., Eisert, J., and Cirac, I. (2019). Expressive power of tensor-network factorizations for probabilistic modeling. In *Advances in Neural Information Processing Systems*, pages 1496–1508.
- Guo, C., Jie, Z., Lu, W., and Poletti, D. (2018). Matrix product operators for sequence-to-sequence learning. *Physical Review E*, 98(4):042114.
- Han, Z.-Y., Wang, J., Fan, H., Wang, L., and Zhang, P. (2018). Unsupervised generative modeling using matrix product states. *Physical Review X*, 8(3):031012.
- Hefny, A., Downey, C., and Gordon, G. J. (2015). Supervised learning for dynamical system learning. In *Advances in neural information processing systems*, pages 1963–1971.
- Heller, A. (1965). On stochastic processes derived from markov chains. *The Annals of Mathematical Statistics*, 36(4):1286–1291.
- Hsu, D., Kakade, S. M., and Zhang, T. (2012). A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480.
- Huggins, W., Patil, P., Mitchell, B., Whaley, K. B., and Stoudenmire, E. M. (2019). Towards quantum machine learning with tensor networks. *Quantum Science and technology*, 4(2):024001.
- Jaeger, H. (1998). Discrete-time, discrete-valued observable operator models: A tutorial. Technical report.
- Jaeger, H. (2000). Observable operator models for discrete stochastic time series. *Neural Computation*, 12(6):1371–1398.
- Kliesch, M., Gross, D., and Eisert, J. (2014). Matrix-product operators and states: Np-hardness and undecidability. *Physical review letters*, 113(16):160503.
- Klümper, A., Schadschneider, A., and Zittartz, J. (1993). Matrix product ground states for one-dimensional spin-1 quantum antiferromagnets. *EPL (Europhysics Letters)*, 24(4):293.
- Kraus, K. (1971). General state changes in quantum theory. *Annals of Physics*, 64(2):311–335.
- Mickelin, O. and Karaman, S. (2018). Tensor ring decomposition. *ArXiv*, abs/1807.02513.
- Miller, J., Rabusseau, G., and Terilla, J. (2021). Tensor networks for probabilistic sequence modeling. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*.
- Monras, A., Beige, A., and Wiesner, K. (2010). Hidden quantum markov models and non-adaptive read-out of many-body states. *arXiv preprint arXiv:1002.2337*.
- Murg, V., Cirac, J., Pirvu, B., and Verstraete, F. (2008). Matrix product operator representations. *New Journal of Physics*, 12:025012.
- Novikov, A., Podoprikin, D., Osokin, A., and Vetrov, D. P. (2015). Tensorizing neural networks. In *Advances in neural information processing systems*, pages 442–450.
- Novikov, A., Rodomanov, A., Osokin, A., and Vetrov, D. (2014). Putting mrfs on a tensor train. In *International Conference on Machine Learning*, pages 811–819.
- Orús, R. (2014). A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158.
- Oseledets, I. (2011). Tensor-train decomposition. *SIAM J. Sci. Comput.*, 33:2295–2317.
- Perez-Garcia, D., Verstraete, F., Wolf, M. M., and Cirac, J. I. (2006). Matrix product state representations. *arXiv preprint quant-ph/0608197*.
- Rabiner, L. and Juang, B. (1986). An introduction to hidden markov models. *IEEE ASSP Magazine*, 3:4–16.
- Schoen, C., Solano, E., Verstraete, F., Cirac, J., and Wolf, M. M. (2005). Sequential generation of entangled multiqubit states. *Physical review letters*, 95 11:110503.
- Siddiqi, S., Boots, B., and Gordon, G. (2010). Reduced-rank hidden markov models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 741–748.
- Singh, S., James, M. R., and Rudary, M. R. (2004). Predictive state representations: A new theory for modeling dynamical systems. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 512–519, Arlington, Virginia, United States. AUAI Press.
- Srinivasan, S., Downey, C., and Boots, B. (2018a). Learning and inference in hilbert space with quantum graphical models. In *Advances in Neural Information Processing Systems*, pages 10338–10347.
- Srinivasan, S., Gordon, G., and Boots, B. (2018b). Learning hidden quantum markov models. In *International Conference on Artificial Intelligence and Statistics*, pages 1979–1987.
- Stokes, J. and Terilla, J. (2019). Probabilistic modeling with matrix product states. *Entropy*, 21(12):1236.

- Stoudenmire, E. and Schwab, D. J. (2016). Supervised learning with tensor networks. In *Advances in Neural Information Processing Systems*, pages 4799–4807.
- Thon, M. and Jaeger, H. (2015). Links between multiplicity automata, observable operator models and predictive state representations — a unified learning framework. *Journal of Machine Learning Research*, 16:103–147.
- Vanderstraeten, L., Haegeman, J., and Verstraete, F. (2019). Tangent-space methods for uniform matrix product states. *SciPost Physics Lecture Notes*.
- Vidyasagar, M. (2011). The complete realization problem for hidden markov models: a survey and some new results. *Mathematics of Control, Signals, and Systems*, 23(1-3):1–65.
- White, S. R. (1992). Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866.
- Wiewiora, E. W. (2008). *Modeling probability distributions with predictive state representations*. PhD thesis, UC San Diego.
- Ying, S. and Ying, M. (2014). Reachability analysis of quantum markov decision processes. *Inf. Comput.*, 263:31–51.
- Yu, R., Zheng, S., Anandkumar, A., and Yue, Y. (2017). Long-term forecasting using higher order tensor rnns. *arXiv: Learning*.
- Zhao, M.-J. and Jaeger, H. (2010). Norm Observable Operator Models. *Neural Computation*, 22(7):1927–1959.