Probabilistic Graphical Models and Tensor Networks: A Hybrid Framework

Jacob Miller[*]
Mila, Université de Montréal
Montréal QC, Canada
jmjacobmiller@gmail.com

Geoffrey Roeder[*]
Princeton University
Princeton, NJ, USA
roeder@princeton.edu

Tai-Danae Bradley
Sandbox@Alphabet
Mountain View, CA, USA
tai.danae@math3ma.com

Abstract

We investigate a correspondence between two formalisms for discrete probabilistic modeling: probabilistic graphical models (PGMs) and tensor networks (TNs), a powerful modeling framework for simulating complex quantum systems. The graphical calculus of PGMs and TNs exhibits many similarities, with discrete undirected graphical models (UGMs) known to be a special case of TNs. However, more general probabilistic TN models such as Born machines (BMs) employ complex-valued hidden states to produce novel forms of correlation among the probabilities. While representing a new modeling resource for capturing structure in discrete probability distributions, this behavior also renders the direct application of standard PGM tools impossible. We aim to bridge this gap by introducing a hybrid PGM-TN formalism that integrates quantum-like correlations into PGM models in a principled manner, using the physically-motivated concept of decoherence. We first prove that applying decoherence to the entirety of a BM model converts it into a discrete UGM, and conversely, that any subgraph of a discrete UGM can be represented as a decohered BM. This method allows a broad family of probabilistic TN models to be encoded as partially decohered BMs, a fact we leverage to combine the complementary benefits of both model families. We experimentally verify the performance of such hybrid models for probabilistic modeling in several real-world datasets, and identify promising uses of our formalism to quantum machine learning and within existing applications of graphical models.

Probabilistic graphical models (PGMs) are a framework for encoding conditional independence information about multivariate distributions as graph-based representations, whose generality and interpretability have made them an indispensable tool for probabilistic modeling. Undirected graphical models (UGMs), also known as Markov random fields, form a general class of PGMs with a diverse range of applications in fields such as computer vision [42], natural language processing [41], and biology [24]. More recently, the graphical structure of discrete UGMs has been shown to be closely related to that of tensor networks (TNs) [34], a state-of-the-art modeling framework first developed for quantum many-body physics [39, 28], whose use in machine learning—for example in model compression [26, 6], proving separations in expressivity between deep and shallow learning methods [8, 20], and as standalone learning models [38, 27]—has been a subject of growing interest.

In this work, we explore the correspondence between UGMs and TNs in the setting of probabilistic modeling. Whereas UGMs are specifically designed to represent probability distributions, general TNs represent high-dimensional tensors whose values can be positive, negative, or even complex. While restricting TN parameters to take on non-negative values results in an exact equivalence with UGMs [34], it also limits their expressivity. More general probabilistic models built from TNs, as

[*]Work done while at Sandbox@Alphabet.

Second Workshop on Quantum Tensor Networks in Machine Learning, 35th Conference on Neural Information Processing Systems (NeurIPS 2021)

exemplified by the Born machine (BM) [17] model family, employ complex latent states that permit them to utilize novel forms of interference phenomena in structuring their learned distributions. While this provides a new resource for probabilistic modeling, it also limits the applicability of foundational PGM concepts such as conditional independence.

We make use of the physics-inspired concept of *decoherence* [45] to develop a hybrid framework for probabilistic modeling, which allows for the coexistence of tools and concepts from UGMs alongside quantum-like interference behavior. We use this framework to define the family of *decohered Born machines* (DBMs), which we prove is sufficiently expressive to reproduce any probability distribution expressible by discrete UGMs or BMs, along with more general families of TN-based models. We further show that DBMs satisfy a conditional independence property relative to its decohered regions, with the operation of decoherence permitting the values of latent random variables to be conditioned on in an identical manner as UGMs. Finally, we verify the empirical benefits of such models using several model architectures and real-world datasets.

Related Work Our work builds on the duality results of [34], which establish a graphical correspondence between discrete UGMs and TNs, by further accounting for the distinct probabilistic behavior of both model classes. Much work across physics, machine learning, stochastic modeling, and automata theory has introduced and explored novel properties of quantum-inspired probabilistic models [44, 2, 11, 14, 32, 31, 17, 37, 36, 3, 5, 23, 13], almost all of which explicitly or implicitly employ tensor networks. The relative expressivity of these models was explored in [16, 1], where quantum-inspired models were proven to be inequivalent to graphical models. Fully-quantum generalizations of various graphical models were investigated in [19].

1 Preliminaries

We work with real and complex finite-dimensional vector spaces $\mathbb{F}^d$, where $\mathbb{F}$ denotes one of $\mathbb{R}$ or $\mathbb{C}$ when the distinction is not needed. We take an nth order tensor, or *n-tensor*, over $\mathbb{F}$ to be a scalar-valued map $T : [d_1] \times \cdots \times [d_n] \to \mathbb{F}$ from an n-fold Cartesian products of index sets, where $[d] := \{1, \dots, d\}$ and where the vector space of all n-tensors is denoted by $\mathbb{F}^{d_1 \times \cdots \times d_n}$. Matrices, vectors, and scalars over $\mathbb{F}$ respectively correspond to 2-tensors, 1-tensors, and 0-tensors, whereas *higher-order tensors* refers to any n-tensor for $n > 2$. The *elements* of T are individual values of T on input tuples, and written as $T_{x_1,\dots,x_n} := T(x_1, \dots, x_n) \in \mathbb{F}$, while the ith *mode* of T refers to the ith argument of T. The *tensor product* of any n-tensor $T \in \mathbb{F}^{d_1 \times \cdots \times d_n}$ and m-tensor $T' \in \mathbb{F}^{d'_1 \times \cdots \times d'_m}$ is the $(n+m)$-tensor $T \otimes T' \in \mathbb{F}^{d_1 \times \cdots \times d_n \times d'_1 \times \cdots \times d'_m}$ whose elements are given by $(T \otimes T')_{x_1,\dots,x_n,x'_1,\dots,x'_m} = T_{x_1,\dots,x_n} T'_{x'_1,\dots,x'_m}$. We use $\mathbb{R}_+$ to indicate the non-negative real numbers, and take the 2-norm of a tensor T to be the scalar $\|T\|_2 = \sqrt{\sum_{x_1,\dots,x_n} |T_{x_1,\dots,x_n}|^2} \in \mathbb{R}_+$. Finally, we use $u^\dagger$ to indicate the conjugate transpose of a complex vector or matrix u.

We focus exclusively on undirected graphs G, whose vertex and edge sets are denoted by V and E. In anticipating the needs of tensor networks, we allow graphs with edges incident to only one node, which we refer to as *visible edges*. We use $E_V \subseteq E$ to indicate the set of all visible edges, and $E_H := E - E_V$ to indicate the set of all *hidden edges*, which are edges adjacent to two nodes. Graphs without visible edges will be called *proper* graphs. For any node $v \in V$, we denote the set of edges incident to v by $\mathrm{Inc}(v)$. A *clique* of G is a maximal subset $C \subseteq V$ such that every pair of nodes $v, v' \in C$ are connected by an edge, and we use $\mathrm{Clq}(G)$ to denote the set of all cliques of G. We define a *cut set* of G to be any set of edges $E_C \subseteq E$ such that the removal of all edges in E_C from G partitions the graph into two disjoint non-empty sub-graphs.

We refer to random variables (RVs) using uppercase letters such as X, Y, Z, and their possible outcomes with lowercase equivalents such as x, y, z. RVs and their outcomes are often indexed with values from an index set, for example $i \in \mathcal{I} = \{1, \dots, n\}$, in which case the notation $X_\mathcal{I}$ indicates the joint RV $(X_1, \dots, X_n)$. A similar notation is used for multivariate functions $f(x_\mathcal{I}) := f(x_1, \dots, x_n)$, as well as for tensor elements $T_{x_\mathcal{I}} := T_{x_1,\dots,x_n}$, and a related notation $\mathbb{F}^{\times_{i \in \mathcal{I}} d_i} := \mathbb{F}^{d_1 \times \cdots \times d_n}$ is used for spaces of tensors. Given three disjoint sets of random variables X_A, X_B, X_C, we let $X_A \perp X_B \,|\, X_C$ indicate the conditional independence of X_A and X_B given X_C, and $X_A \perp X_B$ indicate the (unconditional) independence of X_A and X_B.

1.1 Undirected Graphical Models

Probabilistic graphical models (PGMs) encode multivariate probability distributions using a proper graph $G = (V, E)$ whose nodes v correspond to RVs X_v. We focus on undirected graphical models (UGMs), whose probability distributions are determined by a collection of *clique potentials* $\phi_C : X_C \to \mathbb{R}_+$, non-negative valued functions from the RVs associated with nodes in C, where C ranges over all cliques of G. Given a UGM on an n node graph with clique potentials ϕ_C, the probability distribution encoded by the UGM is

$$P(x_1, \ldots, x_n) = \frac{1}{\mathcal{Z}} \prod_{C \in \mathrm{Clq}(G)} \phi_C(x_C), \qquad (1)$$
$$\text{where } \mathcal{Z} = \sum_{x_1, \ldots, x_n} \prod_{C \in \mathrm{Clq}(G)} \phi_C(x_C).$$

UGMs satisfy a general conditional independence property involving disjoint subsets of nodes $A, B, C \subseteq V$ for which the removal of C leaves the nodes of A and B in separate disconnected subgraphs of G. In this case, the RVs associated with these nodes satisfy $X_A \perp X_B \,|\, X_C$. Such conditional independence guarantees provide a rich and intuitive framework for succinctly encoding the structure of real-world correlations in graphical models, and also help to make the behavior of such models more interpretable.

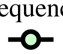
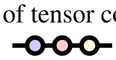
While the definition above is the standard presentation of UGMs, to permit an easier comparison with tensor networks we will more often view them in a dual graphical formulation. In this dual picture, nodes represent clique potentials ϕ_C and edges represent RVs X_i.

2 Tensor Networks

Tensor networks (TNs) provide a general means of efficiently encoding higher-order tensors using smaller tensor cores, in the same manner as UGMs efficiently encode multivariate probability distributions in terms of smaller clique potentials. *Tensor contraction* is the primary computational operation in TNs, and consists of the multiplication of an n-tensor $T \in \mathbb{F}^{d_1 \times \cdots \times d_n}$ and an m-tensor $T' \in \mathbb{F}^{d'_1 \times \cdots \times d'_m}$ along modes k, k' of equal dimension $d_k = d'_{k'}$, to yield a single output $(n + m - 2)$-tensor T'' with elements of

$$T''_{x_1 \cdots x_{k-1} x_{k+1} \cdots x_n x'_1 \cdots x'_{k'-1} x'_{k'+1} \cdots x'_m} = \sum_{x_k=1}^{d_k} T_{x_1 \cdots x_{k-1} x_k x_{k+1} \cdots x_n} T'_{x'_1 \cdots x'_{k'-1} x_k x'_{k'+1} \cdots x'_m}. \qquad (2)$$

Although appearing complex, Equation (2) can readily be seen to generalize matrix-matrix and matrix-vector multiplication, vector inner products, and scalar multiplication. Crucially, tensor contraction is associative, in the sense that iterated contractions between multiple tensors yield the same output regardless of the order of contraction. In practice, finding a good contraction ordering can lead to massive savings in memory and compute when carrying out tensor contraction.

Tensor network diagrams [30] provide an intuitive formalism for reasoning about computations involving tensor contraction using undirected graphs. In a TN diagram, each n-tensor $T \in \mathbb{F}^{d_1 \times d_2 \times \cdots \times d_n}$ is represented as a node of degree n, and each mode of T is represented as an edge incident to T. Tensor contraction between two tensors along a pair of modes is depicted by connecting the corresponding edges of the nodes, with the actual operation of tensor contraction depicted by merging the nodes representing both input tensors into a single node which shares the visible edges of both input nodes. In this manner, a TN diagram with n visible edges and any number of hidden edges specifies a sequence of tensor contractions whose output will always be an n-tensor. For example, the TN diagram  =  expresses a tensor contraction used in the SVD to express a matrix as the product of three smaller matrices. As a special case, the tensor product of two tensors is depicted by drawing them adjacent to each other, with no connected edges.

Tensor networks use a fixed TN diagram to efficiently parameterize a family of higher-order tensors in terms of a family of smaller dense tensor *cores*, as stated in the following:

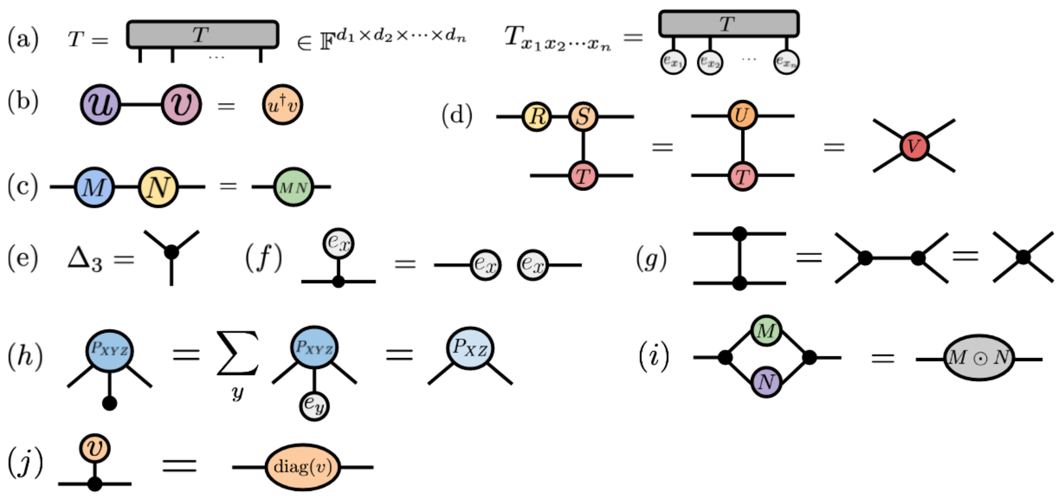
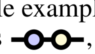


Figure 1: Tensor network and copy tensor notations. (a) Basic notations for TNs, where nodes represent tensors and edges represent tensor modes. Vectors e_x denote elements of an orthonormal basis used to express tensors as arrays. (b–c) Vector inner products $u^\dagger v$ and matrix multiplication MN are simple examples of tensor contraction. (d) Tensor contraction is associative, with the contraction of tensors R, S, and T by first contracting R and S (shown) giving the same result as first contracting S and T. (e) Copy tensors $\Delta_n = \sum_x (e_x)^{\otimes n}$ are denoted by a black dot with n edges. (f) Copy tensors act on basis vectors e_x by copying them to all visible edges and (g) they permit any connected network of copy tensors to be arbitrarily rearranged, provided the total number of visible edges remain unchanged. Copy tensors also allow the graphical expression of basis-dependent operations, including (h) marginalizing over a RV in a probability distribution, (i) the element-wise product of tensors, and (j) the creation of diagonal matrices from a vector of diagonal values.

Definition 1. *A tensor network consists of a graph $G = (V, E)$, along with a positive integer valued map $d_{(-)} : E \to \mathbb{Z}_+$ assigning each edge η to a vector space of dimension d_η, and a map $A^{(-)} : v \mapsto \mathbb{F}^{\times_{\eta \in \mathrm{Inc}(v)} d_\eta}$ assigning each node v to a tensor core $A^{(v)}$ whose shape is a tuple of all dimensions of edges incident to v. The tensor encoded by a tensor network is the contraction of all tensor cores $A^{(v)}$ along the hidden edges of G.*

Dimensions d_i assigned to hidden edges are referred to as *bond dimensions*, and for a fixed graph G they represent the primary hyperparameters setting the tradeoff between a TN's compute/memory efficiency and its expressivity. A simple example of a TN is a low-rank matrix factorization, whose graph G is the line graph on two nodes , and whose single bond dimension is the rank of the parameterized matrix.

Copy Tensors Given an orthonormal basis $\mathcal{B} = \{e_1, \ldots, e_d\}$ for a vector space $\mathbb{F}^d$, for each $n \geq 1$ we define the nth order *copy tensor* associated with $\mathcal{B}$ to be $\Delta_n := \sum_{x=1}^{d} (e_x)^{\otimes n}$. When Δ_n is contracted with any of the d basis vectors e_x, the result is a tensor product $(e_x)^{\otimes n-1}$ of $n-1$ independent copies of e_x (Figure 1f). This convenient property only holds for vectors chosen from the basis defining the copy tensor, leading to a one-to-one correspondence between copy tensor families and orthonormal bases [7]. The copy tensors Δ_1 and Δ_2 respectively correspond to the d-dimensional all-ones vector and identity matrix, with the former allowing the expression of sums over tensor elements.

An nth order copy tensor is depicted graphically as a single black dot with n edges (Figure 1d). Copy tensors satisfy a useful closure property under tensor contraction, with any connected network of copy tensors being identical to a single copy tensor with the same number of visible edges [12, Theorem 6.45]. This property allows connected networks of copy tensors to be rearranged in any manner, so long as the number of visible edges remains unchanged (Figure 1f). General tensor network diagrams can only express basis-independent operations, while the use of copy tensors allows for the graphical description of a larger family of operations which depend on a choice of basis (Figure 1g–i).

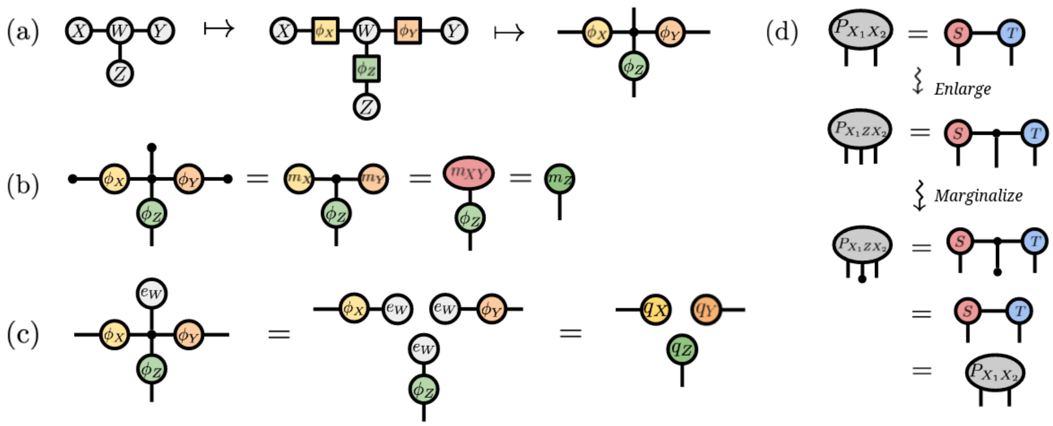


Figure 2: Tensor network description of UGMs. (a) Example of the duality between UGM-style graphical notation, where nodes are associated with RVs, and TN-style graphical notation, where nodes are associated with clique potentials. Factor graphs act as an intermediate representation, with duality simply interchanging variable and factor nodes. (b) Marginalization of a probabilistic model is represented in TN notation by contracting the corresponding visible edges by first-order copy tensors. (c) Conditioning is represented by contracting the corresponding visible edges by outcome-dependent basis vectors, with conditional independence of the resulting distribution achieved through properties of copy tensors. (d) Any TN with non-negative cores can be converted to a UGM by promoting its hidden edges into visible edges by the use of third-order copy tensors. Marginalizing these new latent RVs recovers the original distribution over the visible edges.

2.1 Undirected Graphical Models as Non-Negative Tensor Networks

Discrete multivariate probability distributions are an important example of higher-order tensors, with the individual probabilities $P(x_1, x_2, \ldots, x_n)$ of a distribution over n discrete RVs forming the elements of an n-tensor. More generally any *non-negative tensor* T, whose elements all satisfy $T_{x_1 x_2 \cdots x_n} \geq 0$, can be converted into a probability distribution P_T by normalizing as $P_T(x_1, x_2, \ldots, x_n) = \frac{1}{\mathcal{Z}} T_{x_1 x_2 \cdots x_n}$, where $\mathcal{Z} = \sum_{x_1, x_2, \ldots, x_n} T_{x_1 x_2 \cdots x_n}$.

The connection between multivariate probability distributions and the structure of higher-order tensors extends further, with the independence relation $X_A \perp X_B$ between two disjoint sets of RVs being equivalent to the factorization of their joint probability distribution as the tensor product $P(x_A, x_B) = P(x_A) \otimes P(x_B)$. Methods used to efficiently represent and learn higher-order tensors, such as tensor networks, can be applied to probabilistic modeling, provided that there is some means of parameterizing only non-negative tensors. We discuss two approaches for achieving such a parameterization, one equivalent to undirected graphical models and the other to Born machines.

It was shown in [34, Theorem 2.1] that the data defining a UGM is equivalent to that defining a TN, but with dual graphical notations that interchange the roles of nodes and edges. Converting from a UGM to an equivalent TN involves expressing each clique potential ϕ_C on a clique C of size k as a kth-order tensor core $A^{(v_C)}$, depicted as a degree-k node v_C of the TN diagram. Meanwhile, each UGM node representing a discrete RV is replaced by a copy tensor[1] of degree equal to the number of clique potentials the RV occurs in, plus one additional visible edge permitting the values of the RV to appear in the probability distribution described by the TN (Figure 2a). Since every tensor core consists of non-negative elements, the resultant TN is guaranteed to describe a non-negative higher-order tensor. We refer to this family of TN models as *non-negative tensor networks*.

In the dual graphical notation of TNs, marginalization and conditioning in UGMs comes from contracting a visible edge of the associated TN with a first-order copy tensor $\Delta_1 = \sum_x e_x$ (marginalization) or an outcome-dependent basis vector e_x (conditioning). Computing the distribution over the remaining RVs is then a straightforward application of tensor contraction [34], where any node of the TN with no remaining visible edges is contracted away (Figure 2b). Since variables are associated to copy tensors, the conditional independence property of UGMs can be proven using the copying

[1] Copy tensors were used implicitly in [34], in the form of hyperedges within a defining hypergraph.

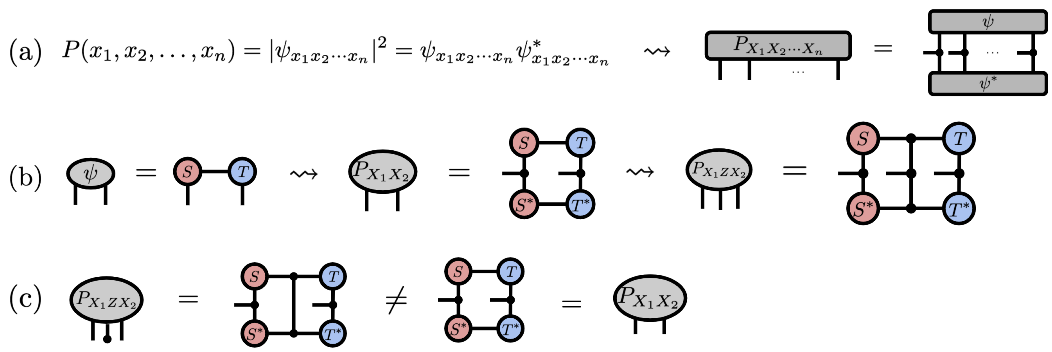


Figure 3: Overview of Born machines, a family of TN-based probabilistic models. (a) Born machines represent a general higher-order tensor ψ as a TN, whose elements are converted to probabilities via the Born rule. This can be used to express the probability distribution itself as a composite TN diagram. (b–c) Unlike non-negative TNs, any attempt to read out the hidden edges of BMs as latent RVs alters the overall distribution, a manifestation of the "observer effect" of quantum mechanics. Converting a hidden edge to a RV and then marginalizing results in a different distribution.

property of copy tensors (Figure 2c). In TN graphical notation, conditional independence arises when a conditioning set of RVs X_C form a cut set of the underlying TN graph, in which case the RVs X_A and X_B associated with the two disconnected subsets of the graph will satisfy $X_A \perp X_B \mid X_C$.

The reverse direction of converting non-negative TNs into UGMs is also straightforward, though care is needed with the treatment of hidden edges. Here we replace all hidden edges by third-order copy tensors, yielding new visible edges encoding latent RVs associated with the previously hidden edges. This process is reversible, in the sense that marginalizing over a latent RV associated with a hidden edge yields the original distribution, allowing hidden edges of a non-negative TN to be freely converted into visible edges (Figure 2d). We will see shortly that this property is not shared by more general probabilistic TN models.

3 Born Machines

While UGMs represent one means of parameterizing non-negative tensors for probabilistic modeling, an alternate approach is suggested by quantum physics. Quantum systems are described by complex-valued *wavefunctions*, higher-order tensors which yield probabilities under the Born rule of quantum mechanics. The efficacy of TNs in learning quantum wavefunctions inspired the Born machine (BM) model [17].

Definition 2. *A Born machine consists of a tensor network over a graph G, whose associated tensor $\psi \in \mathbb{F}^{d_1 \times \cdots \times d_n}$ encodes a probability distribution via the Born rule $P(x_1, \dots, x_n) = \frac{1}{\|\psi\|_2^2} |\psi_{x_1 \cdots x_n}|^2$, where $\|\psi\|_2$ is the 2-norm of ψ and n is the number of visible edges in G.*

The Born rule permits the (unnormalized) probability distribution associated with a BM to be expressed as a single *composite TN*, consisting of two copies of the TN parameterizing ψ, one with all core tensor values complex-conjugated, and where all pairs of visible edges have been merged via third-order copy tensors (Figure 3a). Expressing the BM distribution as a single composite TN allows efficient marginal and conditional inference procedures to be applied in a manner analogous to UGMs, namely by contracting the visible edges of the composite TN with vectors Δ_1 (marginalization) or e_{x_i} (conditioning), and then contracting regions of the TN with no remaining visible edges. The "doubled up" nature of the composite TN means that intermediate states occurring during inference are described by *density matrices*, positive semidefinite matrices employed in quantum mechanics whose non-negative diagonal entries correspond to (unnormalized) probabilities, and whose off-diagonal elements are called "coherences."

The existence of non-zero coherences gives BMs the ability to utilize quantum-like interference phenomena in modeling probability distributions, but also makes it difficult to interpret the operation of a BM by assigning latent RVs to its edges, as was possible with non-negative TNs. While we can force a new RV into existence by extracting the diagonal elements of intermediate density matrices

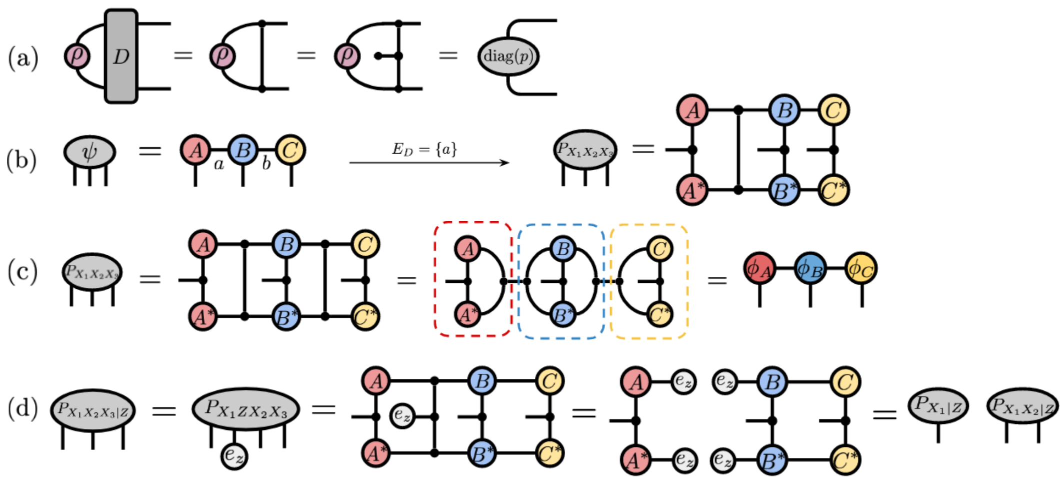


Figure 4: Decoherence and the decohered Born machine (DBM) model. (a) The decoherence operator $\mathcal{D}$, which removes coherences from hidden states in BMs, leaving a diagonal matrix which encodes a latent RV. Decoherence operators permits the readout of latent RVs in a reversible manner, with marginalization of the latent RV yielding the original distribution (third diagram). (b) Examples of decohered Born machines (DBMs) based on a three-core TN with hidden edges a and b. Choosing the decohered edge set $E_D = \{a\}$ results in a decoherence operator being placed in the location corresponding to edge a in the composite TN, which allows the decohered edge to be expressed as a new latent RV. (c) Sketch of the proof of Theorem 1, that every fully-decohered BM is equivalent to a UGM. Copy tensor rewriting rules permit the factorization of fully-decohered BMs into non-negative valued tensors which form clique potentials of an equivalent UGM. (d) Example of the conditional independence property for the DBM above with $E_D = \{a\}$. Conditioning on $Z = z$ for the latent RV at decohered edge a leads to the conditioning value being copied to all attached cores, which in turn leads to a factorization of the conditional distribution into two independent pieces.

using copy tensors (Figure 3b), this causes the elimination of all coherences in density matrices passing through the edge, with the result that the distribution after marginalizing the new latent variable differs from the original BM distribution (Figure 3c). This fact, which can be seen as a consequence of the measurement-induced "observer effect" in quantum mechanics, represents a tradeoff between expressivity and interpretability in general probabilistic TN models that isn't present in PGMs.

4 A Hybrid Framework

While the graphical structure of Born machines can be used to define *area laws*, which characterize the achievable mutual information between subsets of RVs [10, 21], this graphical structure does not directly enable the conditional independence results available with PGMs. The main source of this discrepancy is the difficulty of assigning latent RVs to the hidden edges of a BM, with observations of the visible edges alone unable to guarantee a division of the post-conditioning BM distribution into two independent pieces. While we have seen how latent RVs associated with the hidden edges of a BM can be forced into existence, the fact that this process fundamentally alters the original distribution makes it difficult to easily reason about. However, we show how accounting for this measurement-induced disturbance in a principled manner makes it possible to combine the representational advantages of BMs with the conditional independence guarantees present in PGMs.

We make use of the notion of *decoherence*, a physically-inspired process whereby all off-diagonal coherences of a density matrix are set to zero while leaving all diagonal elements unchanged. Decoherence plays a valuable role in reconciling the counterintuitive nature of measurement in quantum physics with the intuitive behavior of observation in everyday life [45], and we similarly use decoherence to understand the interface between classical and quantum-inspired graphical models. In our setting, we define the *decoherence operator* $\mathcal{D}$ to be simply a fourth-order copy tensor which maps density matrices to other density matrices (Figure 4a). As seen in Figure 3c, $\mathcal{D}$ is the natural result of converting a hidden edge of a BM into a latent RV and then marginalizing. We can use this idea to decohere certain edges of a BM in advance, leading to a division of the hidden edges into decohered and unobserved edges. This naturally leads to the idea of a *decohered Born machine*, which we define as follows:

Definition 3. *A* decohered Born machine *(DBM) consists of a Born machine over a graph G along with an arbitrary subset of "decohered" hidden edges $E_D \subseteq E_H$. The probability distribution represented by a DBM is given by the composite TN associated to the original BM, but with each pair of hidden edges in E_D replaced by a decoherence tensor $\mathcal{D}$. A DBM for which $E_D = E_H$ is referred to as a* fully-decohered *Born machine.*

Having Definition 3 in hand, we would like to understand the expressivity of general DBMs. It is clear that standard BMs are a special case of DBMs, where E_D is the empty set. More generally though, we show in Theorem 1 that taking $E_D = E_H$ leads to a model equivalent to a UGM with the same bond dimensions. This fact, together with Corollary 1, allows us to prove that fully-decohered BMs are entirely equivalent in expressivity to discrete UGMs.

Theorem 1. *The probability distribution expressed by a fully-decohered Born machine with tensor cores $A^{(v)}$ is identical to that of a discrete undirected graphical model with clique potentials of the same shape, whose values are given by $\phi_C(x_C) = |A^{(v)}_{x_C}|^2$, where x_C contains the RVs from all edges adjacent to $v \in V$.*

The proof of Theorem 1 is given in Appendix B.1, with the basic idea illustrated in Figure 4c. Each decoherence operator $\mathcal{D}$ can be written as the product of two third-order copy tensors, each of which can be assigned to one pair of TN cores adjacent to the decohered edge. In the case that all edges of a DBM are decohered, these copy tensors allow each pair of cores $A^{(v)}$ and $A^{(v)*}$ to be replaced by their element-wise product, giving an effective clique potential with non-negative values. The UGM formed by these clique potentials has the same graphical structure as the TN describing the original BM (up to graphical duality). Conversely, the correspondence given in Theorem 1 suggests a basic method for representing any discrete UGM as a fully-decohered BM.

Corollary 1. *The probability distribution of any discrete undirected graphical model with clique potentials $\phi_C(x_C)$ is identical to that of any fully-decohered Born machine with tensor cores of the same shape, and whose elements are given by $A^{(v)}_{x_C} = \exp\left(2\pi i\, \theta_C(x_C)\right) \sqrt{\phi_C(x_C)}$, where θ_C can be any real-valued function, and with $v \in V$ indicating the TN node dual to the clique C.*

Although standard BMs and UGMs operate very differently—and in the case of line graphs have been proven to have inequivalent expressive power [16]—we see that DBMs offer a unified means of representing both families of models with an identical parameterization. Although our above results only characterize the extreme cases of decohering all or none of the hidden edges, we further prove in Appendix B.3 that DBMs are equivalent in expressivity to the class of *locally purified states* [43], a model family which generalizes both BMs and UGMs.

More importantly though, the use of decoherence in DBMs allows the conditional independence guarantees of PGMs to be extended to the setting of quantum-inspired TN models. The ability to replace any decoherence operator by a fifth-order copy tensor with a new visible edge lets us assign RVs to all decohered edges, such that marginalizing over these new RVs yields the original DBM distribution. These new RVs behave identically to those of a UGM, letting us make conditional independence guarantees with respect to the decohered edges of a DBM.

Theorem 2. *Consider a DBM with underlying graph G and decohered edges E_D, along with a subset $E_C \subseteq E_D$ which forms a cut set for G. Denoting the RVs associated to E_C by Z_C, and the RVs associated to the disconnected subsets of G arising from the cut set E_C by X_A and X_B, then the DBM distribution satisfies the conditional independence property $X_A \perp X_B \mid Z_C$.*

While the complete proof of Theorem 2 is given in Appendix C, the idea is simple (Figure 4d). The insertion of decoherence operators, which are examples of copy tensors, into the composite TN for the DBM allows any basis vector e_z used for conditioning to be copied to all edges incident to the copy tensor. This in turn removes any direct correlations between the nodes on either side of the decohered edge, so that conditioning on a collection of RVs associated with a cut set of decohered edges results in a factorization of the post-conditioning composite TN into a tensor product of two independent pieces, and thereby the conditional independence of the associated regions of the DBM distribution.

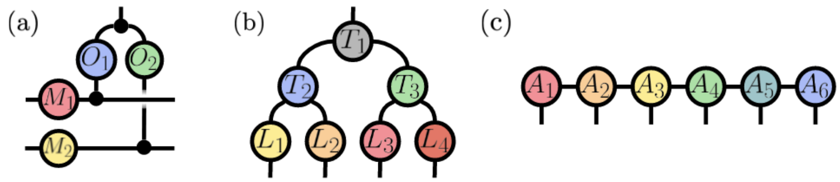


Figure 5: The different underlying TNs used for the experiments. (a) The recurrent cell of a factorial hidden Markov model (FHMM) with two layers. (b) A tree tensor network (TTN) with visible edges at the root and leaf nodes. The TTN used in the congressional voting records experiment contains 16 leaf nodes and 4 layers of hidden edges. (c) A tensor train (TT) model. The TT used in the solar flares experiment contains 13 nodes and 12 hidden edges.

5 Experiments

Table 1: Negative log likelihood (NLL) on held-out test set for the following three experiments: (a) Two-layer factorial hidden Markov model (FHMM) trained on bars and stripes dataset. (b) Tree tensor network model (TTN) trained on congressional voting records dataset. (c) Tensor train model (TT) trained on solar flares dataset. In each case, decohering a strict subset of the model better captures the structure of the data than a pure BM or UGM.

Bars and Stripes w/ FHMM		**Congressional Voting w/ TTN**		**Solar Flares w/ TT**	
None (BM)	36.0	None (BM)	11.6	None (BM)	6.65
One layer	**35.2**	One layer	**11.3**	$^1/_4$ of edges	6.60
All (UGM)	36.7	Two layers	12.8	$^1/_3$ of edges	**6.58**
		Three layers	13.1	$^1/_2$ of edges	**6.58**
		All (UGM)	12.6	All (UGM)	6.63

Having identified several useful theoretical properties of DBMs, we now assess the empirical performance of DBMs as an increasing number of hidden edges of the model are decohered. We use UGM and BM models as natural baselines here, which arise at the limits of decohering all or no hidden edges. We perform experiments on one synthetic and two real-world datasets, which are modeled by DBMs defined over varying graph structures. In each case, we find different levels of decoherence to yield largely similar performance, with models possessing an intermediate amount of decoherence tending to do better than both UGMs and BMs. Our results thus lend support to the idea that decoherence can not only increase the interpretability of quantum-inspired generative models, but also help to better capture the structure present in real-world data.

The first experiment uses an undirected version of a factorial hidden Markov model (FHMM) [15], where the probabilities of several independent HMMs are combined together using product pooling (Figure 5a). We work with a flattened version of the synthetic bars and stripes dataset [22], consisting of 500 8×8 binary images of multiple horizontal or vertical bars, and compare the test loss of the trained model when the hidden states of zero, one, or both HMMs are decohered throughout.

The second experiment uses a tree tensor network (TTN), whose graph is a balanced binary tree and whose visible edges are associated with the root and leaf nodes (Figure 5b). We work with the congressional voting records dataset [35], containing the party affiliation of 435 US representatives along with their votes on 16 different issues. This corresponds to a TTN with 16 leaf nodes and four layers of hidden edges, and we compare the test loss when the topmost k sets of hidden edges are decohered, for $0 \leq k \leq 4$.

The third experiment uses a tensor train (TT) [40, 29], a TN associated to a line graph with a visible edge for each node (Figure 5c). We work with the solar flares dataset [25], containing 1066 records of solar flare events with 13 recorded attributes for each. The associated TT has 12 hidden edges, and we compare the test loss when all, none, or every two, three, or four edges are decohered.

All models in our experiments make use of complex-valued weights, but given that decoherence causes complex phases in the model weights to have no effect on the encoded probability distribution, we choose to increase model bond dimensions as decoherence is increased in each experiment in order to give an approximately equal number of real-valued parameters to each case. We implement all models in JAX [4] and train using adaptive gradient descent with Adam [18]. We divide each dataset into train, validation, and test splits, with the negative log likelihood (NLL) of the model on the test set at the epoch with the lowest validation loss reported in Table 1. Further experimental information, along with our complete experimental code, can be found at `https://github.com/jemisjoky/pgm_tn_bm`.

6 Conclusion

We use the physically-motivated notion of decoherence to define decohered Born machines (DBMs), a new family of probabilistic models that serve as a bridge between PGMs and TNs. As shown in Theorem 1 and Corollary 1, fully decohering a BM gives rise to a UGM, and conversely any subgraph of a UGM can be viewed as the decohered version of some BM. Crucial to this back-and-forth passage is the use of copy tensors, which further allows conditional independence guarantees in the context of TN modeling and provides an additional correspondence between the two modeling frameworks. An immediate limitation of our results surrounding DBMs is the focus on UGMs only. An extension to directed graphical models is left for future work, as is a deeper investigation into what kinds of problems could most benefit from utilizing quantum interference effects in the manner proposed. It is possible that DBMs would improve the performance of existing graphical model inference and learning algorithms by replacing sub-regions of the model with quantum-style ingredients, although a more systematic exploration of this question is needed. The integration of "classical" and "quantum" ingredients represented by a DBM further makes it a natural candidate for quantum machine learning, as decoherence represents a natural form of noise present in quantum hardware in the noisy intermediate-scale quantum (NISQ) era [33].

Acknowledgments and Disclosure of Funding

The authors thank Guillaume Verdon, Antonio Martinez, and Stefan Leichenauer for helpful discussions, and Jae Hyeon Yoo for engineering support. Geoffrey Roeder is supported in part by the National Sciences and Engineering Research Council of Canada (grant no. PGSD3-518716-2018).

References

[1] Sandesh Adhikary, Siddarth Srinivasan, Jacob Miller, Guillaume Rabusseau, and Byron Boots. Quantum tensor networks, stochastic processes, and weighted automata. In *International Conference on Artificial Intelligence and Statistics*, pages 2080–2088. PMLR, 2021.

[2] Raphael Bailly. Quadratic weighted automata: Spectral algorithm and likelihood maximization. In *Asian Conference on Machine Learning*, pages 147–163. PMLR, 2011.

[3] Marcello Benedetti, Delfina Garcia-Pintos, Oscar Perdomo, Vicente Leyton-Ortega, Yunseong Nam, and Alejandro Perdomo-Ortiz. A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Information*, 5(1):1–9, 2019.

[4] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[5] Tai-Danae Bradley, E M Stoudenmire, and John Terilla. Modeling sequences with quantum states: a look under the hood. *Machine Learning: Science and Technology*, 1(3):035008, 2020.

[6] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo P Mandic. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends in Machine Learning*, 9(4-5):249–429, 2016.

[7] Bob Coecke, Dusko Pavlovic, and Jamie Vicary. A new description of orthogonal bases. *Mathematical Structures in Computer Science*, 23(3):555–567, 2013.

[8] Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on Learning Theory*, pages 698–728. PMLR, 2016.

[9] Ralph DeMarr. Nonnegative matrices with nonnegative inverses. *Proceedings of the American Mathematical Society*, 35(1):307–308, 1972.

[10] Jens Eisert, Marcus Cramer, and Martin B Plenio. Colloquium: Area laws for the entanglement entropy. *Reviews of Modern Physics*, 82(1):277, 2010.

[11] Andrew J. Ferris and Guifre Vidal. Perfect sampling with unitary tensor networks. *Phys. Rev. B*, 85:165146, 2012.

[12] Brendan Fong and David I. Spivak. *An Invitation to Applied Category Theory: Seven Sketches in Compositionality*. Cambridge University Press, 2019.

[13] Xun Gao, Eric R Anschuetz, Sheng-Tao Wang, J Ignacio Cirac, and Mikhail D Lukin. Enhancing generative models via quantum correlations. *arXiv preprint arXiv:2101.08354*, 2021.

[14] Xun Gao, Zhengyu Zhang, and Luming Duan. An efficient quantum algorithm for generative machine learning. *arXiv preprint arXiv:1711.02038*, 2017.

[15] Zoubin Ghahramani and Michael I Jordan. Factorial hidden markov models. *Machine learning*, 29(2):245–273, 1997.

[16] Ivan Glasser, Ryan Sweke, Nicola Pancotti, Jens Eisert, and J. Ignacio Cirac. Expressive power of tensor-network factorizations for probabilistic modeling. In *Advances in Neural Information Processing Systems 32*, pages 1496–1508, 2019.

[17] Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised generative modeling using matrix product states. *Phys. Rev. X*, 8:031012, 2018.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Matthew S Leifer and David Poulin. Quantum graphical models and belief propagation. *Annals of Physics*, 323(8):1899–1946, 2008.

[20] Yoav Levine, David Yakira, Nadav Cohen, and Amnon Shashua. Deep learning and quantum entanglement: Fundamental connections with implications to network design. In *International Conference on Learning Representations*, 2018.

[21] Sirui Lu, Márton Kanász-Nagy, Ivan Kukuljan, and J Ignacio Cirac. Tensor networks and efficient descriptions of classical data. *arXiv preprint arXiv:2103.06872*, 2021.

[22] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[23] Jacob Miller, Guillaume Rabusseau, and John Terilla. Tensor networks for probabilistic sequence modeling. In *International Conference on Artificial Intelligence and Statistics*, 2021.

[24] Thierry Mora and William Bialek. Are biological systems poised at criticality? *Journal of Statistical Physics*, 144(2):268–302, 2011.

[25] PM Murphy and DW Aha. UCI repository of machine learning databases (machine-readable data repository). *Inf. Comput. Sci., University of California*, 1995.

[26] Alexander Novikov, Dmitry Podoprikhin, Anton Osokin, and Dmitry Vetrov. Tensorizing neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 442–450, 2015.

[27] Alexander Novikov, Mikhail Trofimov, and Ivan Oseledets. Exponential machines. In *International Conference on Learning Representations*, 2017.

[28] Román Orуś. A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States. *Annals Phys.*, 349:117–158, 2014.

[29] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[30] Roger Penrose. Applications of negative dimensional tensors. *Combinatorial Mathematics and its Applications*, 1:221–244, 1971.

[31] Vasily Pestun, John Terilla, and Yiannis Vlassopoulos. Language as a matrix product state. *arXiv preprint arXiv:1711.01416*, 2017.

[32] Vasily Pestun and Yiannis Vlassopoulos. Tensor network language model. *arXiv preprint arXiv:1710.10248*, 2017.

[33] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.

[34] Elina Robeva and Anna Seigal. Duality of Graphical Models and Tensor Networks. *Information and Inference: A Journal of the IMA*, 8(2):273–288, 06 2018.

[35] Jeffrey Curtis Schlimmer. *Concept acquisition through representational adjustment*. PhD thesis, University of California, Irvine, 1987.

[36] James Stokes and John Terilla. Probabilistic modeling with matrix product states. *Entropy*, 21(12), 2019.

[37] E Miles Stoudenmire. Learning relevant features of data with multi-scale tensor networks. *Quantum Science and Technology*, 3(3):034003, 2018.

[38] EM Stoudenmire and David J Schwab. Supervised learning with tensor networks. *Advances in Neural Information Processing Systems*, pages 4806–4814, 2016.

[39] Frank Verstraete, Valentin Murg, and J Ignacio Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57(2):143–224, 2008.

[40] Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Physical review letters*, 91(14):147902, 2003.

[41] Alex Wang and Kyunghyun Cho. Bert has a mouth, and it must speak: Bert as a markov random field language model. *NAACL HLT 2019*, page 30, 2019.

[42] Chaohui Wang, Nikos Komodakis, and Nikos Paragios. Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 117(11):1610–1627, 2013.

[43] Albert H Werner, Daniel Jaschke, Pietro Silvi, Martin Kliesch, Tommaso Calarco, Jens Eisert, and Simone Montangero. Positive tensor network approach for simulating open quantum many-body systems. *Physical Review Letters*, 116(23):237201, 2016.

[44] Ming-Jie Zhao and Herbert Jaeger. Norm-observable operator models. *Neural Computation*, 22(7):1927–1959, 2010.

[45] Wojciech Hubert Zurek. Decoherence, einselection, and the quantum origins of the classical. *Reviews of Modern Physics*, 75(3):715, 2003.

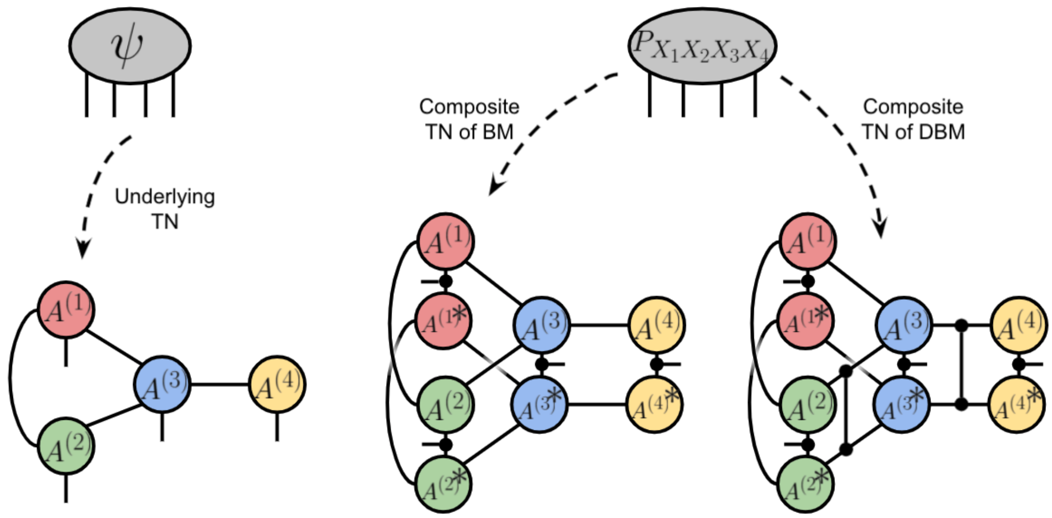


Figure 6: A tensor network (TN) associated with a graph $G = (V, E)$ with $V = \{1, 2, 3, 4\}$ and $E = \{\eta_{1,2}, \eta_{2,3}, \eta_{1,3}, \eta_{3,4}, \eta_1, \eta_2, \eta_3, \eta_4\}$. The edge set E is partitioned into visible edges $E_V = \{\eta_1, \eta_2, \eta_3, \eta_4\}$ and hidden edges $E_H = \{\eta_{1,2}, \eta_{2,3}, \eta_{1,3}, \eta_{3,4}\}$, with the four visible edges giving a fourth-order tensor ψ. A Born machine (BM) uses two copies of this underlying TN, one with all cores complex-conjugated, to define a fourth-order composite TN whose associated tensor $P_{X_1X_2X_3X_4}$ is an unnormalized probability distribution associated with four random variables. A decohered Born machine (DBM) uses a similar composite TN, but allows for a decoherence operator to be inserted in some edges, as specified by a set $E_D \subseteq E_H$. In the composite TN shown, $E_D = \{\eta_{2,3}, \eta_{3,4}\}$.

A Decohered Born Machines Compute Unnormalized Probability Distributions

Here we show that every decohered Born machine (DBM) defines a valid (unnormalized) probability distribution, that is, that the tensor elements of a DBM are non-negative. This can be seen from the fact that the probability distribution represented by a DBM is obtainable as a marginalization of the distribution represented by a larger Born machine (BM) model. By definition, a DBM is a BM ψ with the property that a subset E_D of the set E_H of the hidden edges of ψ are decohered. Recall that decoherence here involves the contraction of k third-order copy tensors Δ_3, where k is the size of the set E_D, and observe that such contraction can be achieved by marginalizing over new latent variables $Z_1, \ldots, Z_k$ introduced in the corresponding hidden edges. The tensor elements of the DBM will then take the form $\sum_{z_1,\ldots,z_k} |\psi'_{x_1,\ldots,x_n,\ldots,z_1,\ldots,z_k}|^2$, where ψ' is associated to a larger BM containing all copy tensors Δ_3 associated with decohered edges. These tensor elements are clearly non-negative, proving that DBMs always describe non-negative tensors.

B Expressivity of Decohered Born Machines

We prove several results which give a general characterization of the expressivity of decohered Born machines (DBMs), showing them to be capable of reproducing a range of classical and quantum-inspired probabilistic models. In Section B.1, we prove that fully-decohered Born machines are equivalent in expressivity to undirected graphical models (UGMs), with the equivalence in question preserving the number of parameters of the two model classes. This result, which parallels the tautological equivalence of non-decohered DBMs and standard BMs, is followed in Section B.3 by a result showing the equivalence of DBMs and locally purified states (LPS), an expressive model class introduced in [16].

We first review the definition of a DBM and some terminology for its graphical structure. Tensor networks (TNs) are defined in terms of a graph $G = (V, E)$ whose edges are allowed to be incident to either two or one nodes in V, and we will refer to the respective disjoint sets of edges as *hidden*

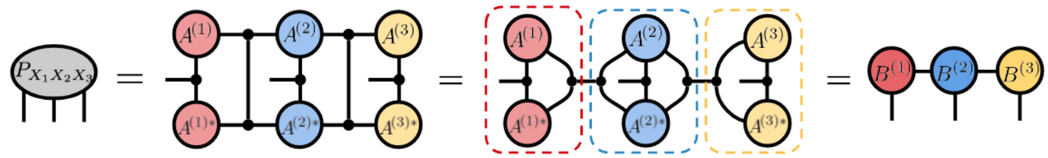


Figure 7: Example of Theorem 1 showing the conversion of a fully-decohered BM into an equivalent UGM. By rewriting each decoherence operator as a product of third-order copy tensors, we can rewrite every pair of BM core tensors $A^{(v)}$ and $A^{(v)*}$ as a single core tensor $B^{(v)}$, whose values are guaranteed to be non-negative. This can consequently be used as a clique potential for a UGM.

edges $E_H \subseteq E$ and *visible edges* $E_V \subseteq E$. Visible edges are associated with the modes of the tensor described by the TN, with the number of visible edges in G equal to the order of the tensor. Nodes which aren't incident to any visible edges are referred to as *hidden nodes* of the TN. Every node $v \in V$ is associated with a tensor *core* $A^{(v)}$ of the TN, with the order of $A^{(v)}$ being equal to the degree of v within G.

Recall that every BM is completely determined by a TN description of a higher-order tensor ψ, whose values are then converted into probabilities via the Born rule. We call the TN describing ψ the *underlying TN*, and the Born rule implies that the probability distribution can be described as a single *composite TN* formed from two copies of the underlying TN, with all pairs of visible edges joined by copy tensors Δ_3. We sometimes use the phrase *composite edge* to refer to any pair of "doubled up" edges in the composite TN, in which case the composite TN can be seen as occupying the same graph as the underlying TN, but where each hidden edge corresponds to a composite edge. The probability distribution defined by a DBM is given by replacing certain composite edges in the composite TN by decoherence operators $\mathcal{D}$, according to whether those edges belong to a set of *decohered edges* E_D (Figure 6).

B.1 Proof of Theorem 1

We first restate Theorem 1, before providing a complete proof.

Theorem 1. The probability distribution expressed by a fully-decohered Born machine with tensor cores $A^{(v)}$ is identical to that of a discrete undirected graphical model with clique potentials of the same shape, whose values are given by $\phi_C(x_C) = |A^{(v)}_{x_C}|^2$, where x_C contains the RVs from all edges adjacent to $v \in V$.

Proof. We show that the composite TN defining the probability distribution of a fully-decohered BM can be rewritten as a TN on the same graph G as the underlying TN, with identical bond dimensions but where all cores take non-negative values. By virtue of the equivalence of non-negative TNs and UGMs [34, Theorem 2.1], this suffices to prove Theorem 1.

Fully-decohered BMs are defined as DBMs for which $E_D = E_H$, so that every composite edge within the composite TN has been replaced by a decoherence operator $\mathcal{D}$. Since $\mathcal{D} = \Delta_4$, we can use the equality of different connected networks of copy tensors (Figure 1g) to express $\mathcal{D}$ as a contraction of two third-order copy tensors Δ_3 along a single edge. Decohered edges are hidden edges and are therefore incident to two distinct (pairs of) nodes in the composite TN. This allows us to move each copy of Δ_3 onto a separate pair of nodes incident to the composite edge (Figure 7). We group together each pair of nodes $A^{(v)}$ and $A^{(v)*}$, along with all copy tensors Δ_3 incident to it, and contract each of these groups into a single tensor, which we call $B^{(v)}$.

It is clear that each tensor $B^{(v)}$ consists of a pair of cores $A^{(v)}$ and $A^{(v)*}$ with all pairs of edges joined together by separate copies of Δ_3. Since this arrangement of copy tensors corresponds to the element-wise product of $A^{(v)}$ and $A^{(v)*}$, this implies that the elements of $B^{(v)}$ satisfy $B^{(v)}_{x_{\mathrm{Inc}(v)}} = \left| A^{(v)}_{x_{\mathrm{Inc}(v)}} \right|^2$, with $x_{\mathrm{Inc}(v)}$ denoting the collection of indices associated with the edges incident to v (these correspond to x_C for some clique C in the dual graph). Since each core $B^{(v)}$ has the same shape as $A^{(v)}$, has

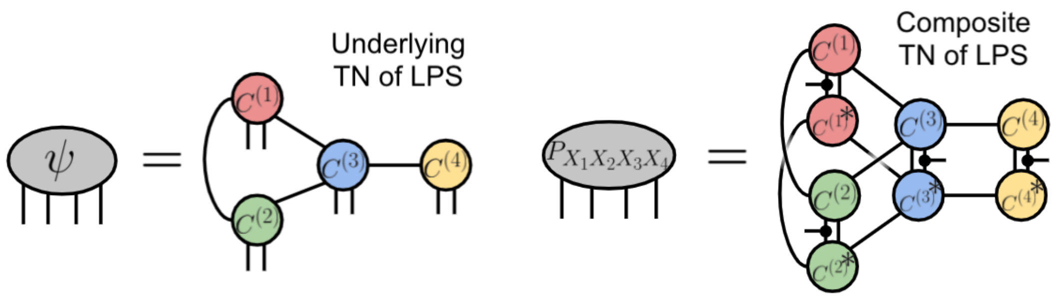


Figure 8: A locally purified state (LPS) model is similar to a BM, but with an additional purification edge added to each node of the underlying TN. Although a small graphical change, this gives LPS greater expressive capabilities than BMs [16]. We show here that DBMs are equivalent to LPS in expressivity.

non-negative values, and is arranged in a TN with the same graph as the underlying TN, this proves Theorem 1.

□

B.2 Proof of Corollary 1

Corollary 1. The probability distribution of any discrete undirected graphical model with clique potentials $\phi_C(x_C)$ is identical to that of any fully-decohered Born machine with tensor cores of the same shape, and whose elements are given by $A^{(v)}_{x_C} = \exp\left(2\pi i\, \theta_C(x_C)\right) \sqrt{\phi_C(x_C)}$, where θ_C can be any real-valued function, and with $v \in V$ indicating the TN node dual to the clique C.

Proof. The statement of Corollary 1 gives an explicit formula for constructing BM cores $A^{(v)}$ from clique potentials ϕ_C, using an arbitrary real-valued tensor θ_C. It can be immediately verified that the conversion from BM cores $A^{(v)}$ to effective clique potentials under full decoherence (Theorem 1) recovers the same clique potentials we had started with, proving Corollary 1. Note that the values of the complex phases $\exp\left(2\pi i\, \theta_C(x_C)\right)$ have no impact on the decohered cores.

□

B.3 Decohered Born Machines are Equivalent to Locally Purified States

Although the definition of locally purified states (LPS) in [16] assumes a one-dimensional line graph for the TN, we give here a natural generalization to LPS defined on more general graphs.

Definition 4. *A locally purified state (LPS) consists of a tensor network over a graph G containing $2n$ visible edges, where all cores contain exactly two visible edges, one of which is designated as a* purification edge*, and the set of purification edges is denoted by $E_P \subseteq E_V$. The n-variable probability distribution defined by an LPS is given by constructing the composite TN for a BM from these cores, with order $2n$, then marginalizing over all n purification edges.*

An illustration of this model family is given in Figure 8. By choosing all purification edges to have dimension 1, LPS reproduce standard BMs, whereas [16, Lemma 3] gives a construction allowing LPS to reproduce probability distributions defined by general UGM. Owing to this expressiveness, and to corresponding results for uniform variants of LPS [1], we can think of LPS as representing the most general family of quantum-inspired probabilistic models. We now prove that DBMs are equivalent in expressivity to LPS, by first showing that LPS can be expressed as DBMs (Theorem 3), and then showing that DBMs can be expressed as LPS (Theorem 4).

Theorem 3. *Consider an LPS whose underlying TN uses a graph $G = (V, E)$ with n nodes, $2n$ visible edges, and m hidden edges. The probability distribution represented by this LPS can be reproduced by a DBM over a graph with $2n$ nodes, n visible edges, and $m + n$ hidden edges, where the decohered edge set E_D is in one-to-one correspondence with the purification edges E_P of the LPS.*

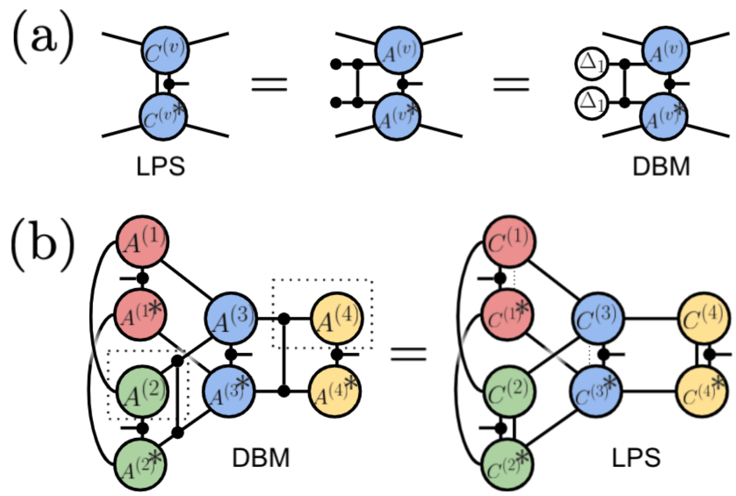


Figure 9: (a) Conversion from an LPS to a DBM. Using diagram rewriting rules, each purification edge joining a pair of LPS cores is expressed as a larger network of copy tensors, which allows the edge to be seen as a decoherence operator $\mathcal{D}$ between the original pair of nodes and a new pair of dummy nodes Δ_1. The result is a DBM associated with an underlying TN with twice as many nodes, and one decohered edge for every purification edge in the LPS (b) Conversion from a DBM to an LPS. In this case, we choose a function f mapping the decohered edges $\eta_{2,3}$ and $\eta_{3,4}$ to nodes 2 and 4, respectively. The dotted boxes show how this can be viewed as defining new cores $C^{(2)}$ and $C^{(4)}$ as the contraction of the DBM cores $A^{(2)}$ and $A^{(4)}$ with adjacent copy tensors Δ_3. The result is an LPS, where we have used dotted edges to indicate trivial purification edges of dimension 1.

Proof. Starting with a given LPS, we construct a TN matching the description in the Theorem statement, whose interpretation as a DBM will recover the desired distribution. We begin with the underlying TN for the LPS, whose n nodes each have one purification edge. We connect each purification edge to a new hidden node, whose associated tensor is the first-order copy tensor Δ_1 with dimension equal to that of the purification edge. This converts all of the purification edges into hidden edges, which form the decohered edges of the DBM.

Given this new TN and choice of decoherence edges, the equivalence of the DBM distribution and the original LPS distribution arises from inserting decoherence operators $\mathcal{D}$ in the composite edges connected to the new hidden nodes, and then using copy tensor rewriting rules to express the composite TN of the DBM as that of the LPS (Figure 9a). Given that the new hidden nodes are associated with constant tensors with no free parameters, and given that all of the cores defining the LPS are kept unchanged in the DBM, the overall parameter count is unchanged. This completes the proof of Theorem 3.

□

Theorem 4. *Consider a DBM defined on a graph $G = (V, E)$ with n nodes and a set of decohered edges $E_D \subseteq E_H$. Given any function $f : E_D \to V$ assigning decohered edges to nodes of G incident to those edges, we can construct an LPS with n nodes which represents the same probability distribution as the DBM. This LPS is defined by a TN with an identical graphical structure to the TN underlying the original DBM, but with the addition of a purification edge at each node v of dimension $d^{(v)} = \prod_{\eta \in f^{-1}(v)} d_\eta$, where d_η is the bond dimension of edge η and $f^{-1}(v)$ is the set of decohered edges mapped to node v.*

Proof. Despite the somewhat complicated formulation of Theorem 4, the idea is simple. In contrast to standard BMs, DBMs and LPS both permit direct vertical edges within the composite TN defining the model's probability distribution, and the proof consists of shifting these vertical edges from decohered edges to the nodes themselves. In the case where multiple vertical edges are moved to a single node, all of these can be merged into one single purification edge by taking the tensor product of the associated vector spaces. This gives the purification dimension $d^{(v)}$ appearing in the Theorem

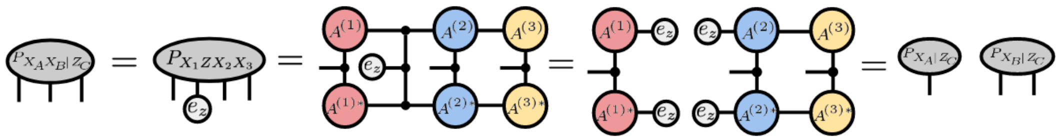


Figure 10: Illustration of the conditional independence of decohered Born machines (DBMs), for a DBM over RV X_1, X_2, X_3, and Z. $Z_C := Z$ is a latent RV associated with the single decohered edge of the DBM, which is a cut set for the underlying graph. Conditioning on the value of z of Z splits the composite TN into two independent pieces, with the result being a probability distribution where $X_A := X_1$ and $X_B := (X_2, X_3)$ are independent RVs.

statement, with the overall procedure illustrated in Figure 9b. For nodes which are not assigned any decohered edges, a trivial purification edge of dimension $d^{(v)} = 1$ is used. This completes the proof of Theorem 4.

□

C Proof of Theorem 2

Theorem 2. Consider a DBM with underlying graph G and decohered edges E_D, along with a subset $E_C \subseteq E_D$ which forms a cut set for G. Denoting the RVs associated to E_C by Z_C, and the RVs associated to the disconnected subsets of G arising from the cut set E_C by X_A and X_B, then the DBM distribution satisfies the conditional independence property $X_A \perp X_B \mid Z_C$.

Proof. From the definition of a cut set, the removal of E_C from the graph for the underlying TN partitions G into two disjoint pieces, and the same property holds true for the composite TN giving the DBM probability distribution. Figure 10 illustrates how conditioning on a decohered edge of a DBM results in the splitting of the associated decoherence operator into a tensor product of two rank-1 matrices, which propagate the value of the conditioning value z to both pairs of incident nodes. Consequently, each composite edge whose value is conditioned on will be removed from the composite TN, and if the set of conditioning edges form a cut set for G, this will result in the separation of the post-conditional composite TN into two disconnected pieces. This implies the independence of the composite random variables X_A and X_B in the conditional distribution, which completes our proof.

□

D Experimental Details

The experiments described in the paper were implemented in JAX [4], and each used a custom implementation of a general DBM model permitting arbitrary underlying TN graphs. All models were trained using stochastic gradient descent with an Adam optimizer [18], and fixed learning rate of 0.001. Each experiment used a fixed number of epochs, which was chosen large enough for the training and validation losses of each model being evaluated to reach a minimum value.

For each experiment, all models being compared share the same graphical structure, with only the decohered edge set E_D and the bond dimensions differing. In general, the bond dimensions of each model were chosen to increase with the number of decohered edges, to keep the number of effective real-valued parameters of the models constant. This decrease in effective parameter count comes from the fact that any fully decohered region of a DBM leads to the decoupling of all complex phases for the associated tensor core parameters from the model probabilities. The effective parameter count dimension of all hidden edges of each model for the three experiments in Section 6 are reported in Table 2.

All code needed for reproducing our experiments can be found at `https://github.com/jemisjoky/pgm_tn_bm`.

Table 2: Bond dimensions of all model configurations, which increased to keep the number of real-valued parameters approximately constant between models.

Decohered Regions	Bond Dim.	# Params
Bars and Stripes w/ FHMM		
None (BM)	5	180
One layer	6	180
All (UGM)	8	192
Congressional Voting w/ TTN		
None (BM)	3	1098
One layer	3	1071
Two layers	3	1017
Three layers	3	909
All (UGM)	4	1136
Solar Flares w/ TT		
None (BM)	10	7980
$1/4$ of edges	10	7935
$1/3$ of edges	10	7935
$1/2$ of edges	10	7935
All (UGM)	14	7770

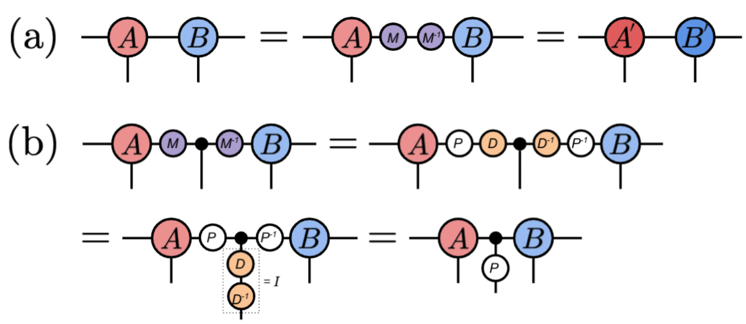


Figure 11: (a) Action of a gauge transformation on a hidden edge of a TN. The insertion of an invertible matrix M and its inverse leads to the adjacent tensor cores A and B being transformed into new tensor cores A' and B', which nonetheless describe the same overall tensor when all hidden edges are contracted together. The use of copy tensors in TNs generally restricts this gauge freedom. (b) The restriction of a TN to have cores with entirely non-negative entries forces any gauge transformation to factorize as $M = PD$, for P a permutation and D a diagonal matrix with strictly positive entries. We show how this restricted gauge freedom mostly commutes with any copy tensor inserted into the hidden edge, with copy tensor rewriting rules allowing us to express such gauge transformations as a trivial permutation of the outcomes of the latent RV associated with the hidden edge. This explains why hidden edges of non-negative TNs can be expressed as latent RVs without loss of generality, allowing a faithful representation as a UGM.

E Gauge Freedom in Probabilistic Tensor Networks

Tensor networks matching the description given in Definition 1 exhibit a form of symmetry in their parameters commonly referred to as *gauge freedom*. This symmetry is generated by edge-dependent *gauge transformations*, wherein an invertible matrix M and its inverse M^{-1} are inserted in a hidden bond of a TN, and then applied to the two tensor cores on nodes incident to that hidden edge. This results in a change in the parameters of the two incident core tensors, which nonetheless leaves the global tensor parameterized by the TN unchanged. The phenomenon of gauge freedom ultimately arises from the close connection between TNs and multilinear algebra, where gauge transformations on a given hidden edge correspond to changes in basis in the vector space associated with the hidden edge.

The use of copy tensors in a TN leads to a preferred choice of basis, and thereby breaks the full gauge freedom of any edge incident to a copy tensor node. It is therefore surprising that for non-negative TNs, i.e. those with all core tensors taking non-negative values, hidden edges were shown to be expressable as latent RVs without loss of generality, via the insertion of copy tensors in the hidden state space (Section 2.1). The generality of this operation means that any non-negative TN can be converted into a UGM by associating hidden edges with latent RVs, where the original distribution over only visible edges is recovered by marginalizing over hidden edges. This fact is a key ingredient in the exact duality between non-negative TNs and UGMs, and differs from quantum-style probabilistic TN models. For example, attempting to observe the latent states associated to hidden edges in a BM will generally lead to a change in the distribution over visible edges, even after marginalizing out these new latent RVs.

We observe here that the generality in associating hidden edges of a non-negative TN to latent RVs is a consequence of the fact that *non-negative TNs already have significantly diminished gauge freedom*. More precisely, in order for a gauge transformation on a hidden edge to maintain the non-negativity of both tensor cores incident to that edge, we must generally have the change of basis matrix M, as well as its inverse M^{-1}, possess only non-negative entries. This is a strong limitation, and is equivalent to the gauge transformation factorizing as a product $M = PD$, where P is a permutation matrix and D is a diagonal matrix with strictly positive entries [9]. We illustrate in Figure 11 how this restricted gauge freedom maintains the overall structure of the copy tensor inserted into a hidden edge, with the result being an irrelevant permutation of the discrete values of the hidden latent variable associated with that edge.

The situation is quite different for BMs and DBMs, and we remark that the use of decoherence operators in a DBM means that the gauge freedom of such models is different than for BMs. In particular, two BMs whose underlying TNs are related by gauge transformations will necessarily define identical distributions, whereas the corresponding DBMs resulting from decohering some gauge-transformed hidden edges may define different distributions. In this sense, the appropriate notion of gauge freedom for a DBM lies in between that of a BM and a UGM defined on the same graph, in a manner set by the pattern of decohered edges.

The choice of basis in which decoherence is performed can be treated as an additional parameter of the model, and we view the interaction between this basis-dependence of decoherence and basis-fixing procedures related to TN *canonical forms* an interesting subject for future investigation.