Mutual information scaling for tensor network machine learning

IOP Publishing Mach. Learn.: Sci. Technol. 3 (2022) 015017 https://doi.org/10.1088/2632-2153/ac44a9

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
29 September 2021

REVISED
16 November 2021

ACCEPTED FOR PUBLICATION
20 December 2021

PUBLISHED
20 January 2022

PAPER

Mutual information scaling for tensor network machine learning

Ian Convy[1,3,*], **William Huggins**[1,3,4], **Haoran Liao**[2,3] and **K Birgitta Whaley**[1,3]

[1] Department of Chemistry, University of California, Berkeley, CA 94720, United States of America
[2] Department of Physics, University of California, Berkeley, CA 94720, United States of America
[3] Berkeley Quantum Information and Computation Center, University of California, Berkeley, CA 94720, United States of America
[4] Present address: Google Quantum AI, Mountain View, CA, United States of America.
[*] Author to whom any correspondence should be addressed.

E-mail: ian_convy@berkeley.edu

Keywords: area law, mutual information, tensor network machine learning

Abstract

Tensor networks have emerged as promising tools for machine learning, inspired by their widespread use as variational ansatze in quantum many-body physics. It is well known that the success of a given tensor network ansatz depends in part on how well it can reproduce the underlying entanglement structure of the target state, with different network designs favoring different scaling patterns. We demonstrate here how a related correlation analysis can be applied to tensor network machine learning, and explore whether classical data possess correlation scaling patterns similar to those found in quantum states, which might indicate the best network to use for a given dataset. We utilize mutual information (MI) as measure of correlations in classical data, and show that it can serve as a lower-bound on the entanglement needed for a probabilistic tensor network classifier. We then develop a logistic regression algorithm to estimate the MI between bipartitions of data features, and verify its accuracy on a set of Gaussian distributions designed to mimic different correlation patterns. Using this algorithm, we characterize the scaling patterns in the Modified National Institute of Standards and Technology and Tiny Images datasets, and find clear evidence of boundary-law scaling in the latter. This quantum-inspired classical analysis offers insight into the design of tensor networks that are best suited for specific learning tasks.

1. Introduction

Tensor decompositions [1, 2], often represented graphically as *tensor networks* [3], have proven to be useful for analyzing and manipulating vectors in very high-dimensional spaces. One area of particular interest has been the application of tensor network methods to quantum many-body physics [4, 5], where the network serves as a parameterized ansatz that can be variationally optimized to find the ground state of a target Hamiltonian [6] or to simulate quantum dynamics [7]. Inspired by these successes in quantum physics, there has been an increased focus in applying tensor networks to machine learning [8–10], where the learning problem is formulated as linear regression on a massively expanded feature space. Although this subset of machine learning research is relatively new, tensor network approaches for classification have already yielded promising results on common benchmark datasets [9, 11, 12].

A central question that arises wherever tensor networks are used, be it in quantum many-body physics or machine learning, is which network structure to choose for a given task. Since matrix product states (MPS) serve as the underlying ansatz for the highly successful density matrix renormalization group (DMRG) algorithm used to calculate ground-state energies [13], researchers in quantum many-body physics have worked to understand the strengths and limitations of these networks. Ultimately, the success of DMRG in 1D systems is made possible by the short-range interactions present in many Hamiltonians, which result in ground states that possess exponentially decaying correlations and localized entanglement that obeys an 'area law' or more properly a *boundary law* [14]. These discoveries have helped motivate the development of other

network structures such as projected entangled pair states (PEPS) [15] and the multiscale entanglement renormalization ansatz (MERA) [16] to deal with multidimensional lattices and quantum critical points respectively.

The purpose of our work is to take the entanglement scaling analysis that has been so illuminating in quantum many-body physics, and adapt it for use on the classical data commonly found in machine learning. Through this analysis, we seek to understand which tensor networks would be most appropriate for specific learning tasks. The body of the paper is organized into five parts: section 2 begins with an overview of tensor networks and their application to machine learning. In section 3 we review how entanglement scaling relates to tensor network methods in quantum many-body physics, and then extend this analysis to classical data by using the mutual information (MI), which provides a generalized measure of correlation. We show that when using tensor networks for probabilistic classification of orthogonal inputs, the MI of the data provides a lower-bound on the entanglement and thus the connectivity of the tensors. Section 4 introduces a numerical method for estimating the MI of a dataset given access to only a finite number of samples. In section 5, we test the accuracy of this method on a set of Gaussian distributions engineered to have different MI scaling patterns with respect to spatial partitioning of the variables. In section 6 we estimate the MI scaling of the Modified National Institute of Standards and Technology (MNIST) images [17] and the Tiny Images [18], two well-known image datasets commonly used in machine learning, and find evidence that the MI between a centered, square patch of pixels and the surrounding pixels scales with the boundary of the inner patch (a boundary law), rather than with the number of pixels (a volume law). This boundary-law scaling suggests that networks with an underlying 2D grid structure such as PEPS would be especially well-suited for machine learning on images.

2. Tensor networks

2.1. Fundamentals

For the purposes of this work, a *tensor* is an array of numbers with a finite number of indices n, each denoted by a distinct subscript. The value of n is called the *order* of the tensor, meaning that vector v_i is a first-order tensor, matrix M_{ij} is a second-order tensor, A_{ijk} is a third-order tensor, and so on. The goal of a tensor *network* is to represent a higher-order tensor as the contraction of a set of lower-order tensors. Since the number of elements in a tensor scales exponentially with the order, a tensor network representation using lower-order tensors can contain exponentially fewer elements than the original tensor, and thus significantly reduce the amount of computational resources required for numerical analysis. For example, a non-cyclic or *open* MPS network (also called a *tensor train decomposition* [19]) represents the nth-order tensor $C_{i_1 i_2 \ldots i_n}$ as the contraction of a sequence of matrices and third-order tensors

$$C_{i_1 i_2 \ldots i_n} = \sum_{\alpha_1 \ldots \alpha_{n-1}} M^{(1)}_{i_1 \alpha_1} A^{(2)}_{\alpha_1 i_2 \alpha_2} A^{(3)}_{\alpha_2 i_3 \alpha_3} \ldots M^{(n)}_{\alpha_{n-1} i_n}, \tag{1}$$

where $i_1, \ldots, i_n$ are the indices corresponding to the higher-order tensor C and $\alpha_1 \ldots \alpha_{n-1}$ are the internal indices of the network that get contracted together. If the indices i_j of C each have dimension d and the internal indices α_j each have dimension m, then the number of elements in the MPS is $\mathcal{O}(dm^2 n)$ while C has d^n elements. If the internal dimension is chosen such that $m \ll d^{\frac{n}{2}}$, then the memory resources needed to represent C are greatly reduced in cases where n is large. This efficiency typically comes at the cost of accuracy, however, since most higher-order tensors cannot be exactly represented by a reasonably-sized MPS or other tensor network, and thus some approximation error is introduced.

When working with tensor networks, it is common to represent expressions such as equation (1) using a graphical notation, where the tensors are represented as geometric shapes and the indices are represented as lines or *legs* protruding outward [15, 20, 21]. The contraction of a pair of indices between two tensors is expressed by connecting the legs of the two tensors together. For example, an open MPS can be expressed graphically as a 1D chain, as shown in figure 1, with the legs of neighboring tensors connected together. A major advantage of the graphical notation is that patterns of connectivity are very clear, even in contractions that involve a large number of tensors. In this paper we augment our tensor network equations with these diagrams to make them easier to visualize.

2.2. Tensor networks for machine learning

The most common forms of discriminative tensor network machine learning can be understood in terms of linear regression, where the model output is generated by a weighted sum of the inputs [22]. This regression

$$\sum_{\alpha_1 \ldots \alpha_4} M^{(1)}_{i_1 \alpha_1} A^{(2)}_{\alpha_1 i_2 \alpha_2} A^{(3)}_{\alpha_2 i_3 \alpha_3} A^{(4)}_{\alpha_3 i_4 \alpha_4} M^{(5)}_{\alpha_4 i_5}$$

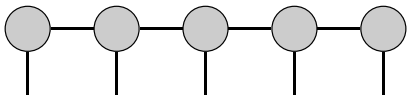


Figure 1. Tensor network diagram for an open MPS containing five tensors, with the corresponding equation given above. The first and last tensors are matrices labeled as M, with the rest being third-order tensors labeled generically as A. The contractions between neighboring tensors are made explicit in the graphical notation, without needing to keep track of specific indices. The order of the MPS is given by the number of uncontracted legs.

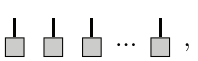
is performed by computing the inner product between a feature vector $\vec{x}$ representing the data and a weight vector $\vec{w}$ representing the model. Taken together with an additive bias b, these produce a scalar output y that lies along a hyperplane

$$y = \sum_i w_i x_i + b. \tag{2}$$

To make these models more expressive, it is common to first transform $\vec{x}$ using a set of feature maps, and then perform the regression. The advantage of such a transformation is that while the output of the model is still linear in the transformed space, it can be a highly non-linear function of the input when mapped back to the original space. A significant drawback is that the outputs of the feature maps may be very high-dimensional and thus too large to store or manipulate. This problem is most often solved using the *kernel trick* [23], where the inner product between feature mappings is used for regression rather than the full feature map vectors. Unfortunately, since the computational cost of many kernel trick methods scales quadratically with the number of samples, this can be impractical for large datasets.

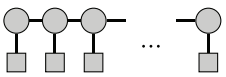
Tensor networks offer a different solution. First, the feature map is constrained to have a tensor product structure

$$X_{i_1 i_2 \ldots i_k} = \bigotimes_{j=1}^{k} f_{i_j}(\vec{x}) \quad = \quad \ldots \quad , \tag{3}$$



where $X_{i_1 i_2 \ldots i_k}$ is the transformed representation of $\vec{x}$, and $\{f_{i_j}\}$ are vector-valued functions transforming the original features. Next, regression is performed directly on the transformed features by fully contracting $X_{i_1 i_2 \ldots i_k}$ with the weight tensor $W_{i_1 i_2 \ldots i_k}$ which is represented by a tensor network. For example, choosing $W_{i_1 i_2 \ldots i_k}$ to be an MPS gives

$$\begin{aligned} y &= \sum_{i_1 \ldots i_k} W_{i_1 \ldots i_k} X_{i_1 \ldots i_k} \\ &= \sum_{i_j, \alpha_j} M^{(1)}_{i_1 \alpha_1} A^{(2)}_{\alpha_1 i_2 \alpha_2} \ldots M^{(k)}_{\alpha_{k-1} i_k} \bigotimes_{j=1}^{k} f_{i_j}(\vec{x}) \\ &= \sum_{i_j, \alpha_j} f_{i_1}(\vec{x}) M^{(1)}_{i_1 \alpha_1} f_{i_2}(\vec{x}) A^{(2)}_{\alpha_1 i_2 \alpha_2} \ldots f_{i_k}(\vec{x}) M^{(k)}_{\alpha_{k-1} i_k} \\ &= \end{aligned} \tag{4}$$



where the last sum can be performed efficiently by contracting the tensors from left to right. For large k a tensor network representation is essential, since the raw weight tensor $W_{i_1 \ldots i_k}$ has far too many elements to operate on directly. However, it is not obvious which type of network to use. Although MPS networks are the most commonly used in the literature, tree tensor networks [24] and MERA have also been employed [25–27]. The wide variety of possible tensor networks raises an obvious question: which structure is best suited for a given machine learning task? In the next section we first describe how the many-body physics community has used the spatial scaling patterns of quantum correlations to answer a similar question when modeling quantum states, and then adapt this analysis for machine learning.

3. Correlation scaling

3.1. Entanglement scaling in quantum systems

Entanglement is a defining property of quantum mechanics [28], and is the source of all correlations between components of a pure-state composite system [29]. Although there are multiple methods of quantifying entanglement, the *entropy of entanglement* is a widely used measure for entanglement between bipartitions of a composite system. For a pure state defined by the joint density matrix ρ_{AB} with reduced density matrices ρ_A and ρ_B corresponding to the bipartitions A and B, the entanglement entropy is defined as the von Neumann entropy of ρ_A (or equivalently, of ρ_B)

$$E(A,B) = -\mathrm{Tr}(\rho_A \log \rho_A). \tag{5}$$

A connection between the entanglement entropy of a quantum state and its structure can be made using the *Schmidt decomposition* [30], which is defined for state $|\psi\rangle$ on the combined Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$ as

$$|\psi\rangle = \sum_{\alpha=1}^{r} \lambda_\alpha |s_\alpha^A\rangle |s_\alpha^B\rangle, \tag{6}$$

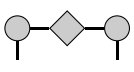
where r is the Schmidt rank, the λ_α are the Schmidt coefficients, and $|s_\alpha^A\rangle, |s_\alpha^B\rangle$ are the orthonormal Schmidt basis states in $\mathcal{H}_A$ and $\mathcal{H}_B$ respectively. Substituting equation (6) into equation (5) gives an expression for the entanglement in terms of the Schmidt coefficients

$$E(A,B) = -\sum_{\alpha=1}^{r} |\lambda_\alpha|^2 \log(|\lambda_\alpha|^2). \tag{7}$$

Formally, the Schmidt decomposition may be regarded as a singular value decomposition (SVD) of the matrix C of coefficients that form $|\psi\rangle$:

$$\begin{aligned} |\psi\rangle &= \sum_{ij} C_{ij} |i_A\rangle |j_B\rangle \\ &= \sum_{ij\alpha_1\alpha_2} V_{i\alpha_1} \Lambda_{\alpha_1,\alpha_2} U^\dagger_{\alpha_2,j} |i_A\rangle |j_B\rangle \\ &= \sum_{ij\alpha} \lambda_\alpha V_{i\alpha} |i_A\rangle U_{j,\alpha} |j_B\rangle \\ &= \sum_{\alpha} \lambda_\alpha |s_\alpha^A\rangle |s_\alpha^B\rangle, \end{aligned} \tag{8}$$

where the rows of C correspond to the computational basis states $|i_A\rangle$ in $\mathcal{H}_A$ and the columns correspond to the computational basis states $|j_B\rangle$ in $\mathcal{H}_B$. The diagonal matrix Λ can be truncated so that it contains only the non-zero singular values of C, which are then equal to the Schmidt coefficients λ_α. Whenever there is more than one non-zero λ_α, the state possesses some degree of entanglement. Since the Schmidt decomposition is an SVD, the set of λ_α is guaranteed to be unique, and the Schmidt rank will be minimized with respect to all possible basis sets. Using the SVD matrices explicitly, we can write the Schmidt decomposition as a small tensor network

$$|\psi\rangle = \sum_{i,j,\alpha_1,\alpha_2} V_{i\alpha_1} \Lambda_{\alpha_1,\alpha_2} U^\dagger_{\alpha_2,j} |i_A\rangle |j_B\rangle \rightarrow$$  , (9)

where V, U are unitary matrices that map the basis states $|i_A\rangle$, $|j_B\rangle$ to the Schmidt bases of $\mathcal{H}_A$ and $\mathcal{H}_B$ respectively. It is important to note that this mathematical description of entanglement, which is based on the singular values, can be used to characterize a tensor regardless of whether it represents a truly quantum object.

The fact that equation (7) arises from a Schmidt decomposition is key to understanding the entanglement scaling properties of tensor networks. As a simple example, the (open) MPS representation of

an N-component quantum system used in algorithms such as DMRG is given by a contraction of second- and third-order tensors, each corresponding to a physical degree of freedom

$$|\psi\rangle = \sum_{i_j,\alpha_j} M^{(1)}_{i_1\alpha_1} A^{(2)}_{\alpha_1 i_2 \alpha_2} \cdots M^{(N)}_{\alpha_{N-1} i_N} |i_1\rangle \cdots |i_N\rangle \rightarrow$$  (10)

If the physical indices are grouped together into two contiguous partitions A and B, with the internal indices contracted within each partition, then equation (10) can be rewritten as

$$|\psi\rangle = \sum_{i,j,\alpha} M^{(A)}_{i\alpha} M^{(B)}_{\alpha j} |i_A\rangle |j_B\rangle \rightarrow$$  , (11)

where i and j are the combined physical indices of partition A and partition B respectively. If the dimension of index α is m, then equation (11) is a canonical decomposition [31] with m terms, having a form similar to that of the SVD in equation (9). Since the SVD, and therefore the Schmidt decomposition, represents the canonical decomposition with the fewest possible terms, the Schmidt rank of an MPS is always upper bounded by m. Through equation (7), this implies that the entanglement entropy represented by an MPS of bond dimension m is bounded by

$$E_{MPS} \leq \log(m), \tag{12}$$

where the inequality is saturated if m is equal to the Schmidt rank and if the singular values are all m^{-1}.

This analysis can be extended beyond MPS [32], with the index α representing the combination of all indices connecting the tensors in the two partitions. Assuming a maximum bond dimension given by m and a number of connecting indices n, the dimension of α is m^n and therefore equation (12) can be extended to a general tensor network as

$$E_{TN} \leq n\log(m). \tag{13}$$

Assuming a fixed bond dimension m, differences in entanglement scaling between tensor networks arise from differences in the value of n, which depends on the geometry of the network. For tensor networks which conform to the physical geometry of the composite system, such as MPS for 1D systems and PEPS for 2D systems, the number of indices connecting two partitions is determined by the size of the interface between the partitions. Given a simple partitioning of the system into a contiguous, hypercubic patch of length L and the surrounding outer patch, the interface scales with the boundary of the inner patch. If the physical lattice dimension is D, the entanglement follows a boundary-law scaling expression

$$E_{TN} \leq 2DL^{D-1}\log(m) = \mathcal{O}(L^{D-1}), \tag{14}$$

where $2DL^{D\text{-}1}$ is the "surface area" of the hypercube. This scaling behavior stands in sharp contrast to that of a random quantum state, whose entanglement will scale with the total size of the inner patch i.e. L^D [33] rather than its boundary in what is sometimes referred to as a 'volume law'. The success of methods like DMRG is only possible because the ground states of common Hamiltonians do not resemble states that have been randomly sampled from the Hilbert space, but instead tend to possess localized, boundary law entanglement that can be readily captured with the MPS ansatz. The existence of such scaling patterns has been proven for the ground states of 1D gapped quantum systems [34], and for harmonic lattice systems of arbitrary dimension [35]. They have also been conjectured to exist in the ground states of most local, gapped quantum systems regardless of dimension [14]. Different tensor networks need to be employed when the ground state is suspected to violate the strict boundary law, with networks such as MERA being used to handle the $\log(L)$ corrections found in many critical-phase Hamiltonians [36]. In any case, the ultimate goal of these tensor network ansatzes is to match the known or predicted entanglement scaling of the quantum state with the entanglement scaling of the network.

3.2. Correlations in classical data

The preceding analysis used entanglement to quantify correlations in a system that was explicitly quantum mechanical. To carry out a similar analysis on classical data, we desire a more general quantity. A reasonable candidate is the MI [37], defined as

$$I(A:B) = S(A) + S(B) - S(AB), \tag{15}$$

where S is the entropy of the probability distributions associated with marginal variables A, B and the joint variable AB. Qualitatively, the MI describes the amount of information we gain about one variable when we learn the state of the other, offering the most general measure of correlation. The MI can be calculated for

either quantum or classical data, depending on whether the von Neumann or Shannon entropies are used. For a pure quantum state $S(AB) = 0$, and therefore the MI is equal to twice the entanglement.

An alternative but equivalent representation of the MI, which we make use of in section 4, comes from the *Kullback-Liebler divergence* (KL-divergence) [38], which is defined for two discrete probability distributions P and Q on variable space $\mathcal{X}$ as

$$D_{KL}(P||Q) = \sum_{x\in\mathcal{X}} P(x)\log\frac{P(x)}{Q(x)}, \tag{16}$$

with an analogous definition for continuous variables that replaces the sum with an integral over probability densities. For a joint probability distribution P over variables A and B in spaces $\mathcal{A}$ and $\mathcal{B}$, the MI is equal to the KL-divergence between the joint distribution $P(A,B)$ and the uncorrelated product-of-marginals distribution $P(A)P(B)$, i.e.

$$I(A:B) = \sum_{a\in\mathcal{A}}\sum_{b\in\mathcal{B}} P(a,b)\log\frac{P(a,b)}{P(a)P(b)}. \tag{17}$$

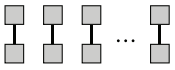
While not formally a metric, the KL-divergence can be viewed as measuring the distance between two distributions, so equation (17) represents the MI as the distance between $P(A,B)$ and the uncorrelated distribution $P(A)P(B)$.

In the context of machine learning, the MI between features in a dataset can be measured by partitioning the features into two groups, assigning the collective state of each group to variables A and B respectively, and then measuring the amount of correlation that exists between the partitions. This parallels the bipartitioning of the quantum many-body system discussed in section 3.1, and allows us to explore *MI scaling* in a similar manner to entanglement scaling.

3.3. Entanglement as a bound on MI for orthogonal data

Given the connection between entanglement and tensor networks discussed in section 3.1, and having introduced the MI as a classical measure of correlation in section 3.2, we now show how the correlations in a classical dataset can guide the choice of network for machine learning. We focus on probabilistic classification, where the tensor network is used to approximate a probability distribution $P(X)$ of feature tensors generated from a classical data distribution $P(\vec{x})$ via equation (3). We show that for orthonormal inputs the entanglement of the tensor network between feature partitions A and B provides an upper bound on the MI of $P(X)$ between those same partitions. When designing a tensor network for a machine learning task, this relationship can be inverted so that the known MI of a given $P(X)$ sets a lower bound on the entanglement needed for the network to represent it. For non-orthogonal inputs these bounds do not hold rigorously, but may still serve as a useful heuristic for samples with negligible overlap.

To begin, let $P(\vec{x})$ be the probability distribution associated with feature vectors $\vec{x}$ of length d corresponding to some set $\mathcal{F}$ of d features. Using a tensor-product map of the form in equation (3), we can map the set of feature vectors $\{\vec{x}\}$ to a set $\mathcal{X}$ of orthogonal rank-one tensors $X \in \mathcal{X}$, generating a new distribution $P(X)$ from $P(\vec{x})$. The overlap of two tensors $X^{(i)}$ and $X^{(j)}$ is determined by the scalar products of the local feature maps

$$\langle X^{(i)}, X^{(j)}\rangle = \prod_{k=1}^{d}\langle f_k(x_k^{(i)}), f_k(x_k^{(j)})\rangle \quad = \quad \text{  }, \tag{18}$$

where each feature map is a function of only a single feature. For this analysis we require that the vectors in the image of each local feature map must form an orthonormal set, so that a pair of feature vectors $\vec{x}^{(i)}$ and $\vec{x}^{(j)}$ will always be mapped to either the same tensor or to a pair of orthogonal tensors. For continuous features, such a mapping can be achieved by discretizing the real numbers into b bins, and then assigning values in each bin to a different b-dimensional basis vector. The f_i for this mapping will never be one-to-one, although as the dimensionality of their outputs grows the functions will come closer to being injective in practice.

Assuming that the images of the local feature maps are finite-dimensional, $\mathcal{X}$ will be finite and therefore $P(X)$ will be a discrete distribution that can be represented as a tensor W of the form

$$W = \sum_{X\in\mathcal{X}} \sqrt{P(X)}X, \tag{19}$$

where we have taken the square-root to ensure that W is normalized (i.e. $\langle W, W\rangle = 1$). With this representation, the probability of a given tensor X can be extracted by taking the square of its scalar product with W

$$P(X) = |\langle X, W\rangle|^2. \tag{20}$$

In the context of machine learning, W can be described using the language of section 2.2 as an idealized weight tensor which we seek to model using a tensor network. For a given network, we want to know which W, and therefore which $P(X)$, can be accurately represented.

To probe the correlations within $P(X)$, we partition the features into disjoint sets $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A}\cap\mathcal{B} = \emptyset$ and $\mathcal{A}\cup\mathcal{B} = \mathcal{F}$. Using this grouping, the underlying feature distribution $P(\vec{x})$ can be represented as the joint distribution $P(\vec{x}_A, \vec{x}_B)$, where $\vec{x}_A$ and $\vec{x}_B$ are vectors containing values for the features in partitions $\mathcal{A}$ and $\mathcal{B}$ respectively. Similarly, $P(X)$ can be represented as the joint distribution $P(X_A, X_B)$, where $\mathcal{X}_\mathcal{A} \ni X_A$ and $\mathcal{X}_\mathcal{B} \ni X_B$ are sets of orthogonal tensors created from the local maps of features in $\mathcal{A}$ and $\mathcal{B}$ respectively. For any tensor $X \in \mathcal{X}$, we have $X = X_A \otimes X_B$ for some X_A and X_B. We can also define the marginal distributions $P(X_A)$ and $P(X_B)$ that describe the statistics within each partition separately. The MI $I(X_A : X_B)$ across the bipartition is given as in equation (15) using the entropies of these distributions.

To introduce the entanglement measure described in section 3.1 as a bound on $I(X_A : X_B)$, we represent the normalized tensor W as the quantum state $|\psi_W\rangle$ and the tensors in $\mathcal{X}$ as orthonormal basis states $|X_A, X_B\rangle$, such that equation (19) becomes

$$|\psi_W\rangle = \sum_{\mathcal{X}_\mathcal{A}, \mathcal{X}_\mathcal{B}} \sqrt{P(X_A, X_B)}|X_A, X_B\rangle, \tag{21}$$

where we have shifted to ket notation. This encoding of a probability distribution into a quantum state has been utilized previously in the study of quantum Bayesian algorithms [39]. The process of extracting $P(X_A, X_B)$ described in equation (20) can be reimagined as projective measurements of $|\psi_W\rangle$ on an orthonormal basis, where the probabilities are used to reconstruct $P(X_A, X_B)$. Since the MI between outcomes of local measurements on a quantum state is upper bounded by the entanglement of that state [40], $|\psi_W\rangle$ must have a bipartite entanglement with respect to partitions $\mathcal{A}, \mathcal{B}$ that is at least as large as $I(X_A : X_B)$. The MI of $P(X)$ across a bipartition therefore provides a lower bound on the amount of entanglement needed in $|\psi_W\rangle$ with respect to that same partition

$$I(X_A : X_B) \leq E(\mathcal{A}, \mathcal{B}), \tag{22}$$

which through equation (13) sets a lower bound on the degree of connectivity n and/or bond dimension m needed in the tensor network representing $|\psi_W\rangle$.

In a typical machine learning setting, we will have access to samples of $P(\vec{x})$, which can then be encoded into tensors which form samples of $P(X)$. If we aim to estimate the MI numerically, as we will in sections 4–6, then it is generally easier to work with the original feature vectors sampled from $P(\vec{x})$ than with the feature tensors from $P(X)$. From the data processing inequality [41], $I(X_A, X_B)$ is upper-bounded by $I(\vec{x}_A, \vec{x}_B)$, so using the MI of the original features will yield a bound on the entanglement that may be larger than necessary to model $P(X)$, but will always be sufficient. Indeed, as the dimensionality of the feature map outputs increases, the gap between $I(X_A, X_B)$ and $I(\vec{x}_A, \vec{x}_B)$ will shrink—since the finer discretization preserves more information—and thus the estimates from both featurizations will converge.

The methodology described above may appear somewhat circuitous, in that we start from the tensorized entanglement formalism that is most natural for tensor networks, but then move back to a classical MI description of the original data features. At first glance it seems like a more direct approach would be to simply estimate the entanglement of $|\psi_W\rangle$ between partitions A and B directly, using some approximation $|\tilde{\psi}_W\rangle$ constructed from the available data

$$|\tilde{\psi}_W\rangle \propto \sum_{i=1}^{N} |X_A^{(i)}, X_B^{(i)}\rangle, \tag{23}$$

where $\{|X_A^{(i)}, X_B^{(i)}\rangle\}$ is a set of N of samples from $P(X_A, X_B)$. Such a construction was recently used for entanglement analysis by Martyn *et al* [42] in the context of MPS image classification. Unfortunately, as evident in [42], the entanglement of $|\tilde{\psi}_W\rangle$ is artificially upper-bounded by $\log(N)$, independent of the actual properties of $P(X_A, X_B)$. This saturation occurs because, for generic sample tensors $|X^{(i)}\rangle$ and $|X^{(j)}\rangle$ with d features, we have

$$\langle X^{(i)}|X^{(j)}\rangle = \prod_{k=1}^{d} \langle f_k(x_k^{(i)}), f_k(x_k^{(j)})\rangle \approx c^d \tag{24}$$

for some typical local overlap $c < 1$. As the number of features grows, the overlap between data tensors is exponentially suppressed. When calculating the entanglement, the near-orthogonality of tensors within $\mathcal{X}_{\mathcal{A}}$ and $\mathcal{X}_{\mathcal{B}}$ (when partitions A and B are both moderately sized) causes the partial trace to generate an almost maximally mixed state with a von Neumann entropy of approximately $\log(N)$. In contrast, by moving back to the original vector space of the data and using MI rather than entanglement, we can generally avoid the $\log(N)$ upper bound (in section 7 we discuss specific circumstances where this limit can also appear in MI estimation).

4. Estimating MI

4.1. Setup and prior work

For our analysis in section 3 to be of practical use, we need a method of estimating the MI of a probability distribution using only a finite number of samples. More formally, let $\{\vec{x}^{(i)}\}$ be a set of N samples drawn from a distribution $P(\vec{x})$ whose functional form we do not, in general, have access to. For a bipartition $\mathcal{A}, \mathcal{B}$ of the dataset features, our goal is to estimate the MI of $P(\vec{x}_A, \vec{x}_B)$ between the features in $\mathcal{A}$ and the features in $\mathcal{B}$ using these samples.

Several approaches to MI estimation [43] have been proposed and explored in the literature. For continuous variables, some methods discretize the variable space into bins, and then compute a discrete entropy value based on the fraction of samples in each bin [44, 45]. Alternatively, kernel density estimators [46] can be used to directly approximate the continuous probability density function using a normalized sum of window functions centered on each sample, which is then used to calculate the MI [47]. A method developed by Kraskov *et al* [48], which utilizes a k-nearest neighbor algorithm to calculate the MI, has become popular due to its improved error cancellation when calculating the MI from approximated entropies.

For this paper, we base our estimation method on more recent work by Koeman and Heskes [49] and Belghazi *et al* [50]. In [49], the MI estimation problem is recast as a binary classification task between samples from $P(\vec{x}_A, \vec{x}_B)$ and $P(\vec{x}_A)P(\vec{x}_B)$, which the authors modeled using a random forest algorithm. In [50], Belghazi *et al* use a neural network to perform unconstrained optimization on the *Donsker-Varadhan representation* (DV-representation) of the KL-divergence between $P(\vec{x}_A, \vec{x}_B)$ and $P(\vec{x}_A)P(\vec{x}_B)$, which provides a lower-bound on the MI. In our work, we found that a mixture of these two approaches was most effective. Specifically, we have used the binary classification framing proposed in [49], but approached the problem as a logistic regression task optimized using maximum log-likelihood on a neural network. To evaluate the MI, we used the DV-representation as in [50] to generate a lower-bound when possible. In practice this also gave us smoother MI curves and smaller errors. To our knowledge this overall approach has not be reported in the literature, though it appears similar in concept to a method proposed by Pool *et al* [51] in the context of generative adversarial networks. In the next subsection we describe our algorithm in more detail.

4.2. Logistic regression for MI estimation

The logistic regression approach to MI estimation is built around the KL-divergence definition of the MI introduced in equation (17). In the context of our dataset, the variable spaces $\mathcal{A}$ and $\mathcal{B}$ describe the collective values of the features in partitions $\mathcal{A}$ and $\mathcal{B}$ respectively, with the sums taken over all allowed value combinations. For convenience, we simplify our notation such that $a \equiv \vec{x}_A$ and $b \equiv \vec{x}_B$ represent the feature values of each partition. To estimate the MI using the KL-divergence, we require an approximation for $f(a,b) = \log \frac{P(a,b)}{P(a)P(b)}$. This can be found via logistic regression by first recasting the joint and marginal probability distributions as conditional probabilities

$$P(a,b|\text{joint}) \equiv P(a,b), \qquad P(a,b|\text{marg}) \equiv P(a)P(b), \tag{25}$$

where $P(a,b|\text{joint})$ is the probability that the feature values a, b will be sampled from the joint distribution $P(a,b)$, and $P(a,b|\text{marg})$ is the probability that the values will be sampled from the product-of-marginals distribution $P(a)P(b)$. Using Bayes' theorem, the conditional probabilities can be reversed

$$P(a,b|\text{joint}) \propto \frac{P(\text{joint}|a,b)}{P(\text{joint})}, \qquad P(a,b|\text{marg}) \propto \frac{P(\text{marg}|a,b)}{P(\text{marg})}. \tag{26}$$

Substituting equations (25) and (26) back into $\log \frac{P(a,b)}{P(a)P(b)}$ gives

$$\log \frac{P(a,b)}{P(a)P(b)} = \log \frac{P(\text{joint}|a,b)}{P(\text{marg}|a,b)} + \log \frac{P(\text{marg})}{P(\text{joint})}, \tag{27}$$

where the first term is the log-odds of a binary classification problem where samples are taken from either $P(a,b)$ or $P(a)P(b)$ and the classifier must decide the most likely source for a given set of feature values a and b. The second term will equal zero if each source is equally likely to be sampled.

To get a numerical estimate of equation (27), we can train a parameterized function $T(a,b)$ to estimate the log-odds via standard logistic regression methods

$$T(a,b) \approx \log \frac{P(\text{joint}|a,b)}{P(\text{marg}|a,b)}, \tag{28}$$

using a training set that consists of an equal number of joint samples and marginal samples. In particular, we parameterized T using a dense feed-forward neural network to avoid introducing spatial bias, and optimized the network by minimizing the binary cross-entropy (i.e. maximizing the log-likelihood) across the samples.

Since the joint distribution is the actual source of our dataset, we already have N samples from it. To approximate a sample from the product-of-marginals distribution, we take a set of values for the features in $\mathcal{A}$ from a joint sample chosen at random, and then take values for the features in $\mathcal{B}$ from another randomly-chosen joint sample (the two sources could be the same sample, although this is unlikely for a large dataset). After selection, the features are combined together into a single mixed sample which, by construction, has no correlations across the partition. After training the network, the MI could be estimated by taking the average of T across the joint samples as a direct approximation[5] of the KL-divergence from equation (17)

$$I(A:B) \approx \frac{1}{M}\sum_{i=1}^{M} T(a_i,b_i), \tag{29}$$

where a_i and b_i are the feature values of the ith joint sample taken from a validation set of size M. However, a superior approach is to insert T into the DV-representation [52] of the MI

$$I(A:B) \geq \frac{1}{M}\sum_{i=1}^{M} T(a_i,b_i) - \log \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} e^{T(a_i,b_j)}, \tag{30}$$

which yields a lower-bound on the MI as $M \to \infty$ and allows errors to cancel[6]. The inequality is saturated when $T = \log \frac{P(a,b)}{P(a)P(b)}$, since as $M \to \infty$ the second term vanishes and the first term gives the KL-divergence. Belghazi *et al* carried out their MI estimation by maximizing equation (30) itself, but we have found in practice that the second term often overflows on datasets with large MI. Furthermore, the optimization algorithm would often attempt to maximize the second term, even though it must vanish in the optimal solution. We were able to mitigate these problems by instead training with the binary cross-entropy [53] as a loss function and only using equation (30) at the end to get the MI value of the optimized distribution. As a caveat, we found in practice that for certain distributions with larger MI values equation (29) generally yielded more stable and accurate estimates than equation (30), though the reason for this is not clear.

5. Numerical tests with Gaussian fields

5.1. Gaussian Markov random fields

To test the accuracy of the logistic regression algorithm, we need a distribution to sample from that has an analytic expression for the MI and that can model different MI scaling patterns. Both of these requirements are satisfied by Gaussian Markov random fields (GMRFs) [54], which are multivariate Gaussian distributions parameterized by the *precision matrix* $Q \equiv \Sigma^{-1}$, where Σ is the more familiar covariance matrix. With respect to Q, the Gaussian distribution with mean $\vec{\mu}$ is

$$p(\vec{x}) = \sqrt{\det\left(\frac{1}{2\pi}Q\right)}\exp\left[-\frac{1}{2}(\vec{x}-\vec{\mu})^T Q(\vec{x}-\vec{\mu})\right], \tag{31}$$

where $p(\vec{x})$ is the probability density of the variables $\vec{x}$. The element Q_{ij} of the precision matrix determines the *conditional correlation* between variables x_i and x_j, which describes the statistical dependence of the pair when all other variables are held fixed at known values. This is in contrast with the more familiar *marginal*

[5] For M samples of $P(a,b)$, we have $\lim_{M\to\infty} \frac{1}{M}\sum_{i=1}^{M} T(a_i,b_i) = \sum_{a\in\mathcal{A},b\in\mathcal{B}} P(a,b)T(a,b)$.

[6] For example, errors of the form $T(a,b)+\epsilon$ can be brought outside of the sums in equation (29) to give $\epsilon - \log e^{\epsilon}$, which cancels.

correlation, governed by Σ, which describes the dependence between a pair of variables when the state of all other variables is unknown. If $Q_{ij} = 0$, the variables x_i and x_j are conditionally uncorrelated:

$$p(x_i, x_j | x_{k\neq i,j}) = p(x_i | x_{k\neq i,j}) p(x_j | x_{k\neq i,j}) \iff Q_{ij} = 0. \tag{32}$$

By setting specific elements of the precision matrix to zero, the correlation structure and therefore the MI of the Gaussian can be tuned to a desired pattern. This flexibility allows us to encode different MI scaling patterns into the distribution, which can then be extracted analytically using equation (15) and the expression for Gaussian entropy

$$S = \frac{1}{2} \log[\det(2\pi e \Sigma)]. \tag{33}$$

Substituting equation (33) into equation (15) gives an expression for the Gaussian MI:

$$I(A:B) = \frac{1}{2} \log \frac{\det(\Sigma_A) \det(\Sigma_B)}{\det(\Sigma)}, \tag{34}$$

where Σ_A and Σ_B are the covariance matrices corresponding to variables in partitions A and B respectively.

5.2. Test setup

In the following subsections, we present test results of the logistic regression estimator on GMRFs representing three different correlation patterns: a boundary law with nearest-neighbor correlations, a volume law with weak correlations across all variables, and a distribution with sparse, randomized correlations. In the language of quantum many-body physics, the first two patterns reflect correlation structures that would be expected in ground states and random states respectively, while the GMRF with random sparse correlations shows the scaling for a heterogeneous distribution that lacks any spatial structure. These Gaussian distributions serve as both a means of testing the algorithm and as a clear illustration of the numerical MI plots that would be expected from different types of correlations within a dataset.

In the tests, each GMRF consisted of 784 variables, which mirrored the number of pixels in the 28×28 images taken from the MNIST and Tiny Images datasets analyzed in section 6. To measure the scaling behavior of the MI in these GMRFs, we used a range of different bipartition sizes, with the partitions being selected such that they always formed a pair of contiguous patches when the variables were arranged in a 28×28 array. One member of each bipartition was formed from an inner square patch of variables centered on the array, whose side length we denote as L. The other partition was an outer patch consisting of all other variables. The size of the inner partition ranged from a single variable ($L = 1$) to a 26×26 block ($L = 26$). For each bipartition, the MI was estimated using our logistic regression algorithm and the DV-representation, with the estimates plotted alongside the analytic MI curve of the GMRF to evaluate their quantitative and qualitative accuracy. Since our model used a stochastic gradient descent method for optimization, we averaged over multiple training runs to generate a representative curve. To explore the effect of sample size on the algorithm, we generated datasets from the GMRFs with 70 000, 700 000, and 7000 000 joint training samples and created MI curves for each size using averages over 20, 10, and 5 trials respectively. Samples and covariance plots from the GMRF test distributions are given in appendix A2.

5.3. Nearest-neighbor boundary-law GMRF

As shown in equation (14), for the MI of a bipartition to obey a boundary law its magnitude must scale with the length of the boundary or interface between the partitions. Given a set of variables on a d-dimensional lattice, the simplest way to construct a boundary law is to have each variable be conditionally-correlated with only its $2d$ nearest neighbors. For variable x_{ij} on a two-dimensional grid at row i and column j, the conditional probability function would depend on the values of only four other variables

$$p(x_{i,j} | \{x_{k\neq i,j}\}) = p(x_{i,j} | x_{i+1,j}, x_{i-1,j}, x_{i,j+1}, x_{i,j-1}), \tag{35}$$

although the number of neighbors can be fewer if the variable is at an edge or corner since the grid is finite. After partitioning, the inner patch of variables will be conditionally correlated with only a single layer of variables surrounding its perimeter, so the MI between the inner and outer partitions will be proportional to L. To encode the correlation structure of equation (35) into a precision matrix, all of the off-diagonal elements in each row of Q must be set to zero except those that correspond to the nearest neighbors, with the

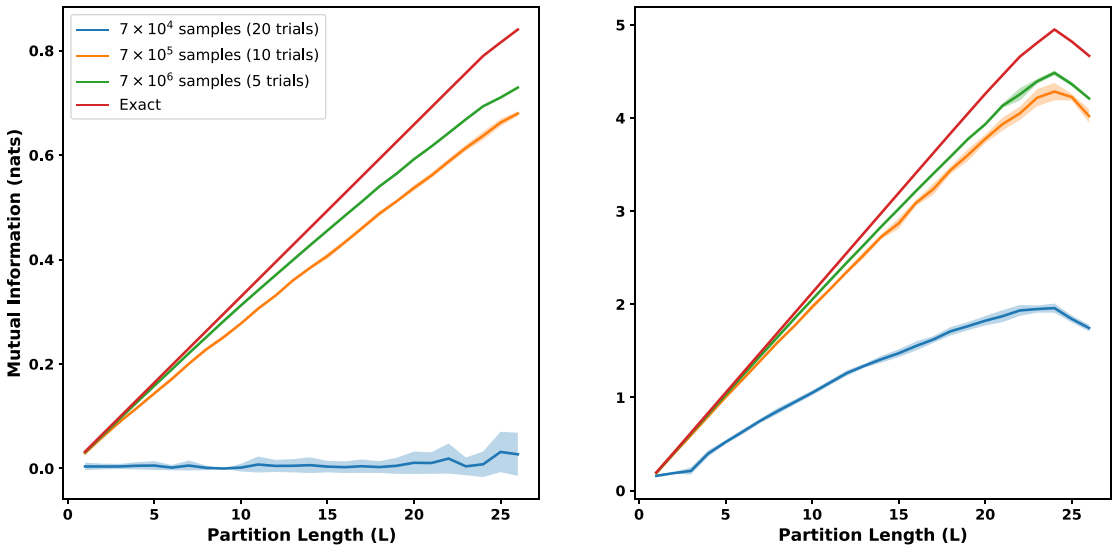


Figure 2. MI curves for a GMRF with nearest-neighbor correlations at different sample sizes, plotted relative to the side length L of the inner partition. The plot on the left had a weaker correlation strength with $q = -0.12$, while the plot on the right had a stronger correlation strength with $q = -0.227$. The solid lines represent the averages over the trials, while the shaded regions show one standard deviation. The linear boundary-law scaling pattern of the GMRF is evident from the exact curve (red). With the exception of the weakly-correlated, 70 000 sample trial, this linear scaling is successfully reconstructed by the algorithm. The magnitude of the MI is underestimated in all trials, with the fractional error being similar for the strong and weak correlations.

non-zero off-diagonal elements all assigned the same value q that determines the strength of the correlation. To guarantee that Q is positive definite, q should not exceed the magnitude of the diagonal elements divided by the number of nearest neighbors (see section 5.4 for more details on these constraints).

The performance of the logistic regression algorithm on the nearest-neighbor GMRF is summarized in figure 2, which plots the MI in *nats* against the side length L of the square inner partition[7]. Since the x-axis is proportional to the perimeter of the inner patch rather than its area, we expect a boundary-law MI curve to be linear in L. This is clearly evident in the analytic curve, which is linear up to a length of roughly 25 variables before leveling off. The linear pattern is broken near the boundaries because the marginal correlations between variables around the edges of the grid are smaller than those between variables closer to the center. Aside from the the 70 000 sample trial with weak correlations, the regression estimates were able to successfully reproduce the boundary-law scaling pattern, with the error shrinking as the number of samples increased. It is also interesting to note that the fractional errors of the different sample sizes are similar between the strong and weak correlations, suggesting that the source of the error is independent of the MI magnitude.

5.4. Uniform volume law GMRF

In contrast with the local correlations that give rise to a boundary law, we can imagine an alternative pattern in which each variable is equally correlated with every other variable. These correlations produce a volume law for the MI, since every variable in the inner partition must contribute equally to the correlations with the outer partition. To encode such a pattern into a GMRF, we set every off-diagonal element of the precision matrix Q to the same value q. To ensure that the precision matrix remains positive definite, the value q should be small enough to preserve *diagonal dominance*, a sufficient but not strictly necessary condition for a positive definite matrix in which the sum of the magnitudes of the off-diagonal elements of a row or column do not exceed the diagonal element

$$Q_{ii} > \sum_{i \neq j} |Q_{ij}| \text{ and } Q_{ii} > \sum_{j \neq i} |Q_{ij}|. \tag{36}$$

To create a uniform scaling pattern it suffices to set $Q_{ii} = 1$, which means we must have $q < \frac{1}{N-1}$ for an N-dimensional Gaussian. This provides an upper limit on the amount of correlation one Gaussian variable

[7] When calculating quantities such as the entropy or MI using natural logarithms, the unit of information is a *nat* instead of a *bit*.

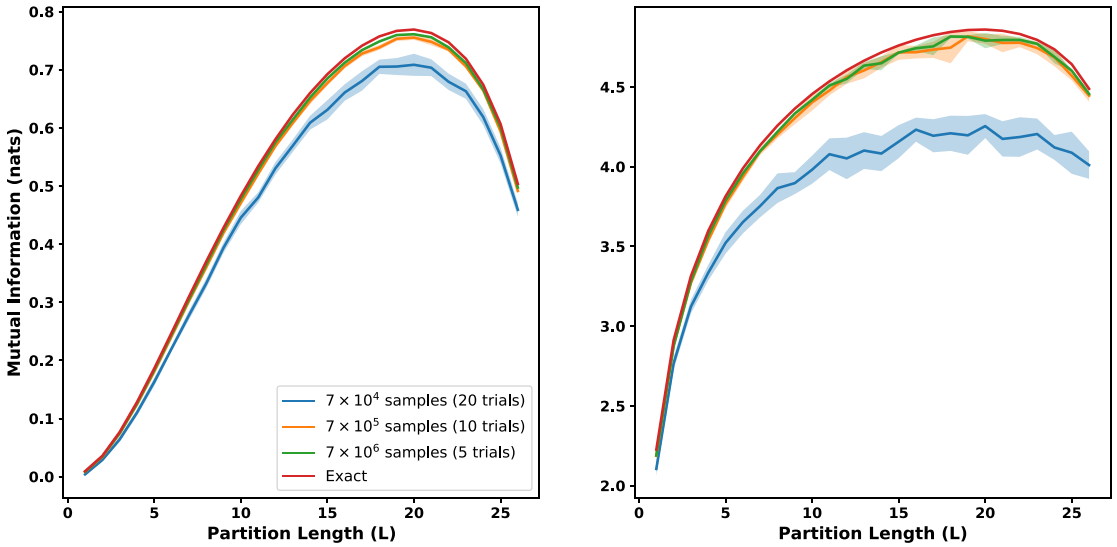


Figure 3. MI curves for a GMRF with uniform correlations at different sample sizes, plotted relative to the side length of the inner partition. The plot on the left had a weaker correlation strength with $q = -1.2 \times 10^{-3}$, while the plot on the right had a stronger correlation strength with $q = -1.27712 \times 10^{-3} \approx \frac{1}{783}$. The solid lines represent the averages over the trials, while the shaded regions show one standard deviation. The algorithm successfully reproduced the shape of the exact MI curve, with the larger sample sizes almost matching the analytic MI values. The finite size of the grid causes the curve to gradually bend over as the partition length increases.

and can have with any other when the correlations are homogeneous, a limit that decreases as the number of variables grows larger.

The performance of our algorithm on a GMRF with these uniform correlations is summarized in figure 3. We were able to accurately reproduce the shape and approximate magnitude of the analytic curves for both correlation strengths and for all sample sizes, although as expected the 70 000 sample trials had the largest error. Interestingly, the algorithm performed significantly better on the uniform GMRF than on the nearest-neighbor GMRF, even though the pairwise dependence between correlated variables in the former was much weaker than in the latter. This suggests that, for a given amount of MI, it is easier for the algorithm to find correlations that are spread out across many variables than to identify those that are concentrated in some sparse set.

It is worth noting that the shape of a volume-law curve should be quadratic on the axes used in figure 3, yet from our plots it is clear that the quadratic form breaks down quickly for the weak correlations and never exists at all for the strong correlations. This distortion occurs because the MI is purely a function of the number of variables in each partition when the correlations are homogeneous, and due to the finite size of our grid any increase in the size of the inner patch necessarily comes at the cost of the outer patch. Correspondingly, any increase in the MI that comes from growing the inner patch is partially offset by the correlations that are lost when shrinking the outer patch. On the 28×28 grid used in figure 3, the MI begins to decline at partition length $L = 20$, which marks the point where both partitions contain roughly the same number of variables (400 vs 384) and where the amount of correlation is therefore maximized.

5.5. Random sparse GMRF

A third class of GMRF to explore is one where the correlations have no inherent spatial pattern yet are also non-uniform. Such a distribution could, for example, represent a dataset of features that are correlated but lack the in-built sense of position or ordering necessary to unambiguously map them onto a lattice (e.g. demographic data). If we nevertheless insist on embedding these features into a grid, we can expect that for most arrangements the MI will scale either as a volume law or in some irregular pattern, depending on whether the features all have similar correlation strengths.

For our tests, we engineered a spatially-disordered GMRF by taking the nearest-neighbor precision matrix used in section 5.3 and randomly permuting the variables around the grid. Under this scheme, each row and column of the precision matrix Q has four non-zero off-diagonal elements in random positions. While the conditional correlations of this new distribution are still sparse, they are no longer exclusively short-range but can instead span the entire grid. Since all of the non-zero off-diagonal elements of Q have the same magnitude q, the amount of correlation across any bipartition increases evenly with the number of

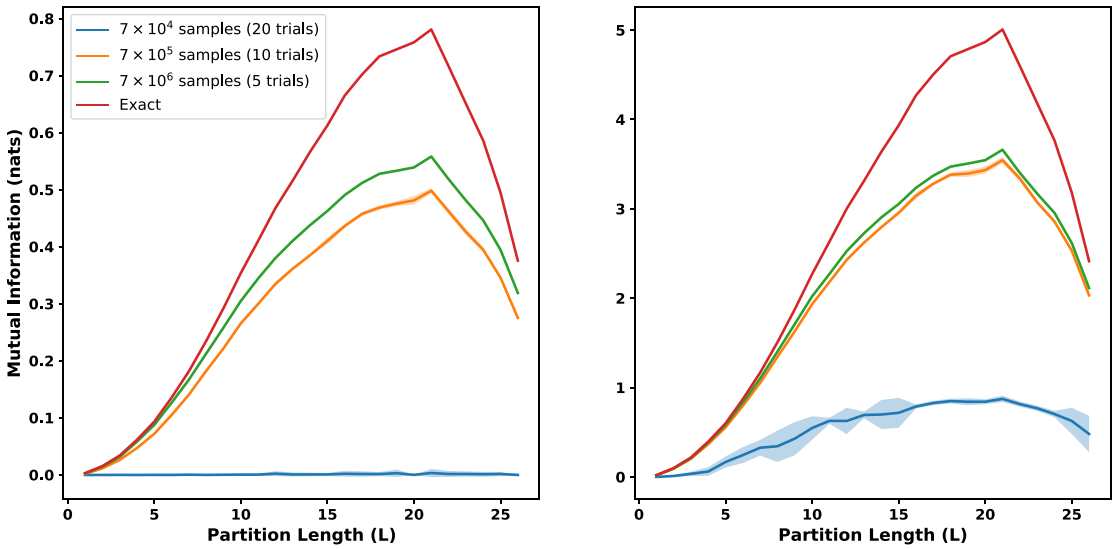


Figure 4. MI curves for a GMRF with spatially-randomized correlations at different sample sizes, plotted relative to the side length of the inner partition. The plot on the left had a weaker correlation strength with $q = -0.045$, while the plot on the right had a stronger correlation strength with $q = -0.11$. The solid lines represent the averages over the trials, while the shaded regions show one standard deviation. The quadratic, volume-law scaling is clear from the analytic MI curve at small L, which for the strong correlations was reproduced across all sample sizes (though the 70 000 sample curve is greatly diminished). For the weak correlations the model was unable to find any correlations using the smallest sample size of 70 000, as was the case for the nearest-neighbor correlations (figure 2, left panel).

correlated variable pairs shared between the partitions. Without any underlying spatial structure, the odds of a given pair being separated into two different partitions is roughly proportional to the volume of the smaller partition, assuming that the other partition is much larger. Under the inner-outer partitioning scheme used in our tests, we expect a volume law for small partition lengths, followed by the same bending-over observed in section 5.4 for the uniform correlations.

The performance of our logistic regression algorithm on a GMRF with these spatially-randomized correlations is shown in figure 4. As predicted, the analytic curves show similar scaling patterns to those of the uniform GMRF in figure 3. The quality of the MI estimates, however, is more similar to the nearest-neighbor MI curves of figure 2, where the model succeeded at replicating the analytic MI curve for all sample sizes when the correlation strength was large, but failed for the smallest sample size (70 000) when the correlations were weak. The estimation error is larger overall for the randomized variables than for the nearest-neighbor variables, and increasing the sample sizes appears to yield diminishing returns. This may partially stem from the reduction in pairwise correlation strength (quantified by q) that was required to keep the magnitude of the peak MI value consistent between the different GMRFs. However, a more likely explanation is that the nearest-neighbor correlations are able to reinforce one another due to their shared proximity, which results in the next-nearest-neighbor marginal correlations also being quite strong. This may make the correlations easier for a machine learning algorithm to detect, since they will impact a larger number of variables. In contrast, for the randomized GMRF the correlated variables are scattered far away from each other on average, which severely diminishes any reinforcement effect.

6. Application to image data

6.1. Setup

To explore the types of MI scaling patterns that might be seen in real data, we analyzed two sets of images: the 70 000 image MNIST handwritten-digits dataset [17], and 700 000 images taken from the Tiny Images dataset [18] converted to grayscale using a weighted luminance coding[8]. Sample images from these datasets and further details can be found in appendix A1. These two datasets were chosen due to their differing levels of complexity: MNIST consists of simple, high-contrast shapes while the Tiny Images are low-resolution depictions of the real world with much more subtle color gradients. In our experiments, each image

[8] For normalized RGB color values, each grayscale pixel was assigned the value 0.3 R + 0.59G + 0.11B. See [55] for more information on grayscale conversions.

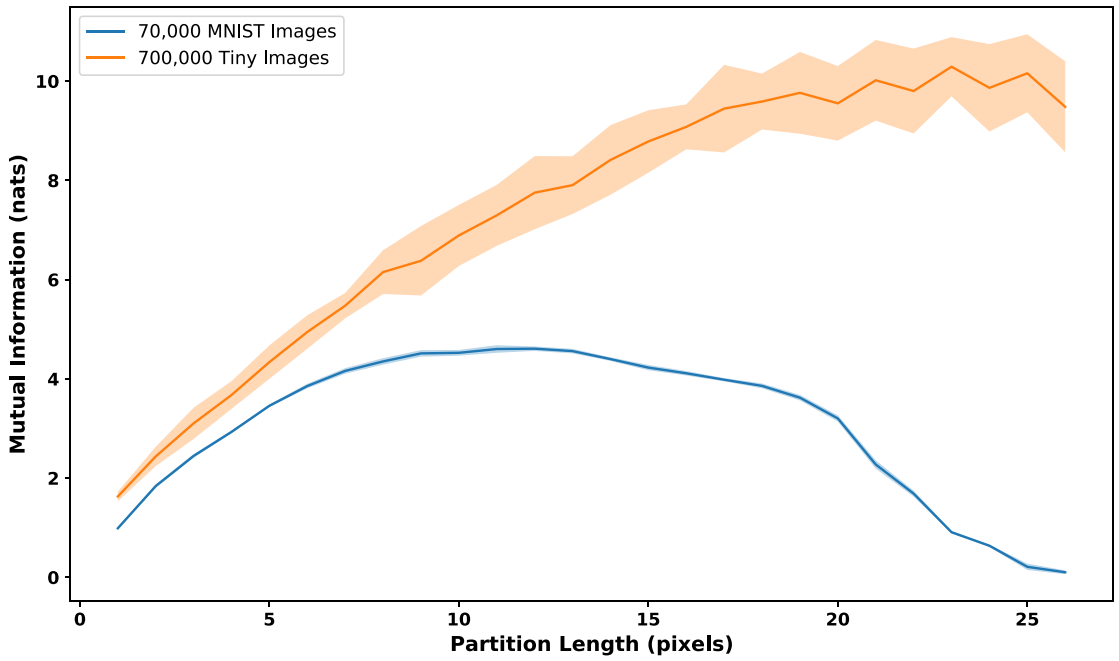


Figure 5. MI estimates for the MNIST and Tiny Images datasets using logistic regression, plotted relative to the side length L of the inner partition. The solid lines are averages from 20 separate trials, while the shaded regions show one standard deviation. The MNIST curve most closely resembles the strongly-correlated uniform GMRF from section 5.4, and exhibits minimal variance. The Tiny Images curve is most similar to the nearest-neighbor, boundary-law GMRF from section 5.3 (figure 2, right panel), but the shape is harder to pin down due to its high variance.

contained 784 pixels arranged in a 28×28 array, with the Tiny Images dataset being cropped from 32×32 by removing two pixels from each side. The pixel values, originally integers from 0 to 255, were rescaled to the range $[0, 1]$.

To generate the MI estimates for these two datasets, we used the same partitioning method described in section 5 for the GMRFs, with each image being split into a centered, square inner patch of increasing size and a surrounding outer patch. These partitions were then fed into the algorithm laid out in section 4, with one key difference; the DV-representation of equation (30) proved to be unusable for both MNIST and the Tiny Images due to instability in the exponential term. While we were able to use the DV-representation to significantly reduce error on the GMRF tests, on the real datasets we had to instead make a direct estimate of the KL-divergence from equation (29). It is not clear why the DV-representation worked for the GMRFs but not for the image datasets, although this could be due to the larger MI and stronger correlations that are present in the real-world data.

6.2. Results

Figure 5 shows the MI of the MNIST and Tiny Images datasets as estimated by logistic regression, plotted relative to the side length L of the inner pixel partition. The MI curves were generated from averages taken over 20 different trials, and plotted within a shaded region containing one standard deviation. As with the GMRFs, this averaging helped smooth the curves and make their shapes easier to assess, especially for patch sizes with larger variance.

Looking first at the Tiny Images curve, we can see a moderately linear segment from 1 pixel length to roughly 18 pixels length, which then flattens out and begins to decrease at the 26×26 patch. Of the three scaling curves tested in section 5, this overall shape is most consistent with the boundary-law scaling pattern of section 5.3 (figure 2, right panel). Unfortunately the variance of the algorithm increased significantly at larger MI values, making it more difficult to assess the pattern. For MNIST, the MI curve most closely resembles that of the strongly-correlated uniform GMRF (figure 3), rising at a decreasing rate until it crests and gradually declines. However, this shape is not as distinct as that of a linear or quadratic curve, so it is difficult to use as evidence for a volume law.

Interestingly, the MNIST curve shows far less variance than the Tiny Images curve, despite the fact that it contains only a tenth of the images. For the GRMF tests done in section 5, there was a clear reduction in the variance of each curve as the sample size increased, but this not observed in figure 5. Indeed, the MNIST

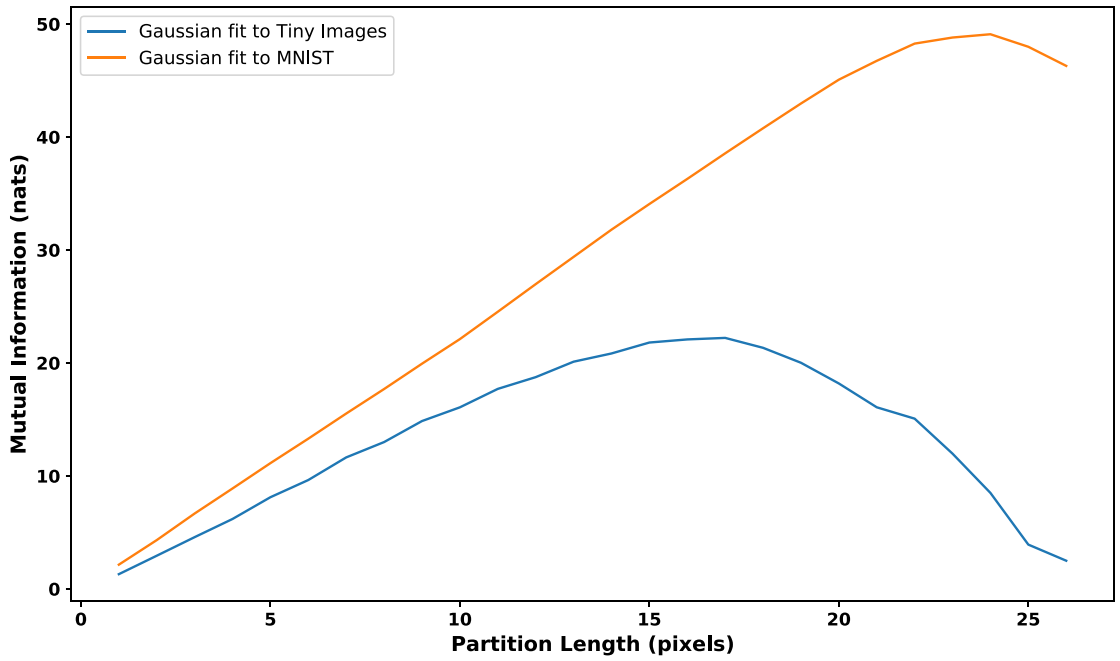


Figure 6. Analytic MI curves from the GMRFs fitted to MNIST and the Tiny Images, plotted relative to the side length L of the inner partition. The Tiny Images curve shows a clear boundary law, while the MNIST curve also starts linear but gradually bends over. Since the GMRFs only model simple pairwise correlations, these MI values are very likely underestimates.

curve has a smaller variance at each patch size than the Tiny Images curve has at almost any patch size, even when the MI of the MNIST curve is larger. This suggests that there is some data-specific effect causing the discrepancy, perhaps attributable to the relative simplicity of the MNIST images relative to the more realistic Tiny Images.

Unlike in our GMRF tests, we do not have access to the underlying probability distributions that MNIST and the Tiny Images datasets were sampled from, so it is much more difficult to assess the accuracy of the curves in figure 5. One approximate way of evaluating the estimates is to fit a GMRF to the empirical covariance matrix of the data, and then calculate the Gaussian MI analytically in the same manner as in section 5. This new distribution is constrained to model only pairwise interactions between the variables, and all marginal and conditional distributions among the variables are forced to be Gaussian, so it is not representative of the true distribution. Nevertheless, due to its high entropy and simple correlation structure, a fitted GMRF is likely (but not guaranteed [56]) to provide a lower bound on the MI of the true distribution.

Figure 6 shows the Gaussian MI curves generated by fitting GMRFS to the covariance matrices of both the MNIST and Tiny Images datasets. It is important to note the scale of the y-axis: the MI values obtained from the fitted GMRFs are roughly five times larger than the predictions of the logistic regression algorithm that are shown in figure 5, indicating a severe underestimation in the latter. The curve for the Tiny Images in figure 6 is remarkably linear, only declining at the end because of the finite size of the image. This agrees with the shape of the logistic regression curve in figure 5 and almost exactly resembles the boundary-law GMRF curve from section 5.3 (figure 2). The MNIST GMRF curve is also approximately linear up to an inner patch length of roughly $L = 15$ pixels, at which point the curve bends over and begins to decrease due to finite size effects; these are exacerbated by the fixed black border placed around each digit[9]. While the MNIST curves in figures 5 and 6 have somewhat similar shapes at larger patch sizes, the linearity of the Gaussian MNIST curve in figure 6 at small L is not present in the corresponding regression curve of figure 5. Taken together, these results show that if the GMRF estimates are viewed as approximations of the simple, pairwise correlations in the images, then it is evident that the scaling behavior of those correlations obeys a clear boundary law in both datasets. Samples and covariance plots from the two fitted GMRFs are given in figure 10.

[9] The MNIST digits themselves are only 20×20 pixels in size; since the digits are roughly centered in the 28×28 image, most of the outer pixels on the edges will be uniformly black and thus contribute nothing to the MI.

Although the primary focus of this work is the use of logistic regression as a means of quantifying MI scaling, it is clear from figure 6 that GMRF techniques offer a viable alternative. We provide here a brief discussion of the relative merits of each method. Compared to a stochastically-optimized neural network, a multivariate Gaussian is very simple to fit and provides a single, deterministic MI estimate via equation (34). The logistic regression algorithm, by contrast, shows significant variation across trials even when the dataset is fixed, a problem which becomes more severe at larger MI values (see, e.g. the Tiny Images plot in figure 5). The simplicity of the GMRF comes at a cost, however, since Gaussians are inherently quadratic and thus incapable of modeling interactions between more than two variables. We would expect complex datasets to posses these higher-order dependencies, which favors the use of more expressive neural network models. At the same time, we can see from comparing figure 5 with figure 6 that the logistic regression method captures only a fraction of the total magnitude of the MI. Collectively, these observations suggest that the GMRF approach should be favored when the correlation patterns are simple or when only a rough lower-bound on the MI of a dataset is desired. By contrast, regression with a neural network is better suited to estimate the MI of data with more complex correlations.

7. Discussion

Recent work in quantum many-body physics has shown that the success of a tensor network ansatz is closely tied to the correlation structure of the underlying system. It stands to reason that similar logic should hold in machine learning. If true, this presents us with two main challenges. First, on a theoretical level, we must gain insight into the mathematical relationships that exist between dataset correlations and network architecture. At the same time, on a more practical level, we need to be able to quantify and characterize the kinds of correlation structures present in real-world data. Our work here addresses both of these problems, using the classical MI to establish an entanglement lower-bound for probabilistic classification tasks and finding clear evidence for boundary-law scaling in the Tiny Images dataset.

On the theoretical side, we established in section 3.3 that the MI of the data features provides a lower bound on the entanglement needed for probabilistic classification of orthogonal samples by a tensor network. We showed that direct entanglement estimates, taken from the state representing the sample distribution, are artificially upper-bounded by the logarithm of the number of samples, regardless of the nature of the distribution. When the true entanglement is expected to exceed this bound, such as for data with a large number of features, a different measure of correlation such as the MI is therefore necessary. Given that the entanglement of a network with fixed bond dimension is $n \log m$ (equation (13)), an MI estimate can help determine both the connectivity of the network (n) and the size of the indices (m). While the lower bound should still hold approximately on samples with small overlaps, it will be useful to explore in future work whether and to what extent it is possible to generalize this bound to non-orthogonal featurizations. Additionally, there are many machine learning tasks where the ground truth cannot be expressed as a probability or modulus—e.g. regression over the real numbers $\mathbb{R}$—and which therefore fall outside of our analysis. It seems likely that the correlation structures in these tasks would still be important when choosing the right tensor network, but the mathematical relationship is not as clear as in the probabilistic cases studied here.

Assuming that the images analyzed in section 6 can be mapped to tensors with minimal overlap and that therefore the bound in section 3.3 applies, then our numerical results suggest that the MI of the Tiny Images obeys a boundary law. The evidence is less definitive for MNIST, although the analytic curve obtained by fitting a GRMF shows a clear boundary law for smaller patch sizes. This would indicate that the most appropriate tensor network to use for probabilistic classification of these datasets from a correlation standpoint is PEPS, whose connectivity follows a 2D grid. However, given that exact contraction of a large PEPS network is impossible even with small bond dimension, it would be useful to look at alternative structures that still possess a 2D geometry. Some possibilities include a TTN with four child nodes, or networks with a Cayley tree structure [57] possessing four nearest neighbors.

From a numerical perspective, our present work on MI estimation appears to be one of the few in the literature that seeks to quantify the spatial structure of the MI, or even just approximate the magnitude of the MI itself. Instead, most of the existing research focuses on MI as a minimization or maximization target, as seen in various independent component analysis algorithms [58] or in the training of generative models [59]. To our knowledge, the only other work that explores MI scaling is that of Cheng *et al* [60], which

characterized the MI of MNIST in the context of training sparse Boltzmann machines. The authors utilized side-to-side and checkerboard partitioning schemes, focusing their analysis on the degree to which the estimated MI value (using Kraskov's nearest-neighbor method) differed from the maximum MI value that could exist between the partitions. While their results showed that the estimate was significantly smaller than the maximum, it is unclear how much of this was actually an intrinsic property of the data or just a numerical limitation of the nearest-neighbor method used for estimation.

Indeed, recent work by McAllester and Stratos [61] has shown that lower-bound MI estimates based on sampling, such as our logistic regression algorithm using the DV representation, can never produce an estimate greater than $\mathcal{O}(\log N)$, where N is the number of samples. If we make the reasonable assumption that the Gaussian curves from figure 6 underestimate the true MI, then we would need on the order of 10^{21} images to get a good estimate of the Tiny Images MI. This is of course impossible. For MNIST, the number of samples needed is on the order of 10^8, which is within the realm of possibility but would require a massive data collection and training scheme. On a practical level, this means that the DV representation cannot be used for MI estimation on datasets that have strong correlations, although it is unclear whether the $\log(N)$ bound tells us anything about direct approximations of the KL divergence in the spirit of equation (29) (which was used to produce figure 5). McAllester and Stratos recommend instead to minimize the cross-entropy as an upper bound on the entropy, then use equation (15) to get an estimate of the MI that is not a lower bound. This could be a useful direction for future work.

Tensor network machine learning is still in its infancy, and there is much work to be done in understanding the strengths and weaknesses of different network designs. It is likely that dataset correlations present in a given task will dictate the tensor structure that is best suited for the job, but determining which correlations are most important, and knowing how to assess that importance, is challenging. We have shown here that the scaling of the MI within a dataset can be systematically characterized in a manner that parallels the entanglement scaling analysis performed on quantum states, which may provide insight into these questions.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Acknowledgments

I C was supported by the US Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Quantum Algorithm Teams Program, under Contract Number DE-AC02-05CH11231. W H was supported by Siemens #FutureMakers Fellowship 045695. H L was supported by the National Aeronautics and Space Administration under Grant/Contract/Agreement No. 80NSSC19K1123 issued through the Aeronautics Research Mission Directorate. Computational resources were provided by the Molecular Graphics and Computation Facility at the University of California, Berkeley (NIH Grant S10OD023532).

Appendix

A1. MNIST and tiny images datasets

MNIST and the Tiny Images datasets (figure 7) are common benchmarks used in computer vision research. MNIST consists of 70 000 samples of handwritten digits collected from high school students and Census Bureau employees by the National Institute of Standards and Technology (NIST) and further processed by LeCun, Cortes, and Burges. The original NIST images were bilevel, with each pixel represented by a single bit as either black or white. Grayscale shades were then introduced incidentally when the digits were reshaped to fit in a 28×28 array.

The Tiny Images dataset is a set of approximately 80 million images collected by Torralba, Fergus, and Freeman. The dataset was gathered from the internet by searching for 75 062 nouns using a variety of search engines. The images were downsampled to 32×32 pixels, with each pixel represented as a vector in RGB color space. For a better comparison to MNIST, we converted the colored images to grayscale and cropped them to down to a size of 28×28.

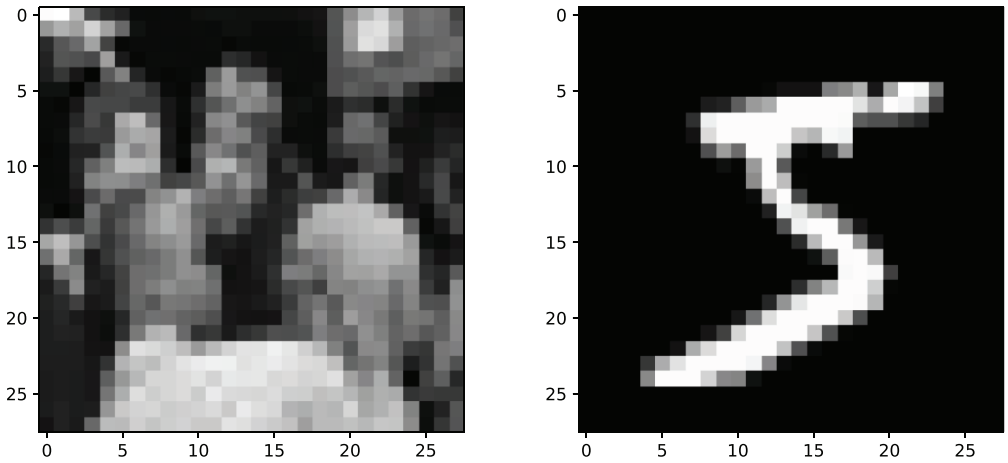


Figure 7. Left: low-resolution image taken from the Tiny Images dataset after having been cropped and converted to grayscale. Right: the handwritten digit '5' taken from MNIST. The axes give the height and width of each image.

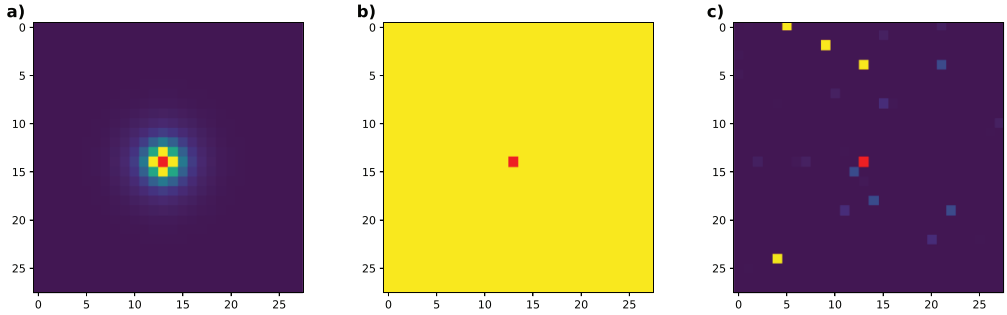


Figure 8. Covariance plots for the (a) nearest-neighbor, (b) uniform, and (c) randomized GMRF distributions used in section 5. The covariance values are taken with respect to the center variable highlighted in red, with brighter colors indicating stronger correlations and darker pixels indicating weaker correlations.

A2. GMRF covariances and sample images

Figure 8 shows covariance plots of the three GMRFs tested in section 5 with respect to a single variable highlighted in red. The magnitudes are expressed as colors to emphasize the importance of the correlation pattern rather than the specific covariance values. The variables that have the strongest covariance with the center variable are bright yellow, and correspond to the variables which have a non-zero conditional correlation with the center variable. In figure 8(a) the four nearest-neighbor variables are clearly visible, while in figure 8(c) those four variables are randomly distributed throughout the image. In figure 8(b) the covariance matrix is uniformly yellow, as every variable is conditionally correlated with every other variable. Samples from these GRMFs are shown in figure 9, where the subtly of the correlation effects is evident.

The covariance plots and sample images shown in figure 10 are taken from the GRMFs fit to the Tiny Images and MNIST. The samples posses considerably more structure than those in figure 9, which is consistent with the large MI values found in figure 6. That said, the GRMFs are clearly not able to capture the full structure of the underlying dataset distributions, since the Tiny Images GMRF does not resemble any identifiable object and the MNIST GMRF sample does not resemble any digit. The covariance plots of figure 10 both show strong nearest-neighbor correlations, which is consistent with the boundary-law scaling observed in figure 6.

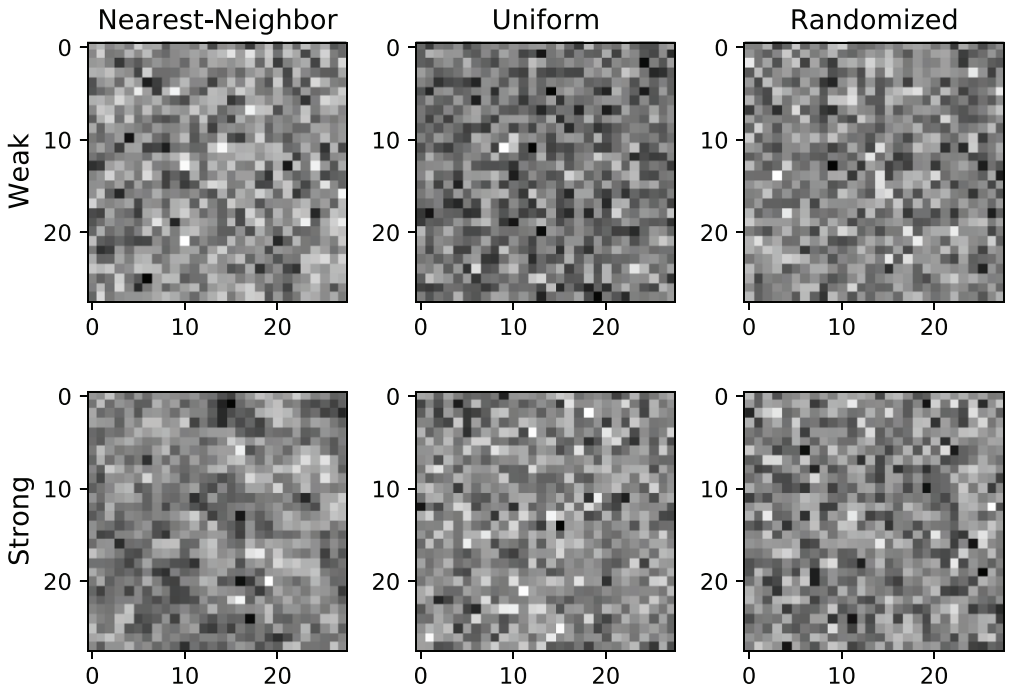


Figure 9. Sample 'images' taken from the GRMF distributions of section 5 at both strong and weak correlation strengths.

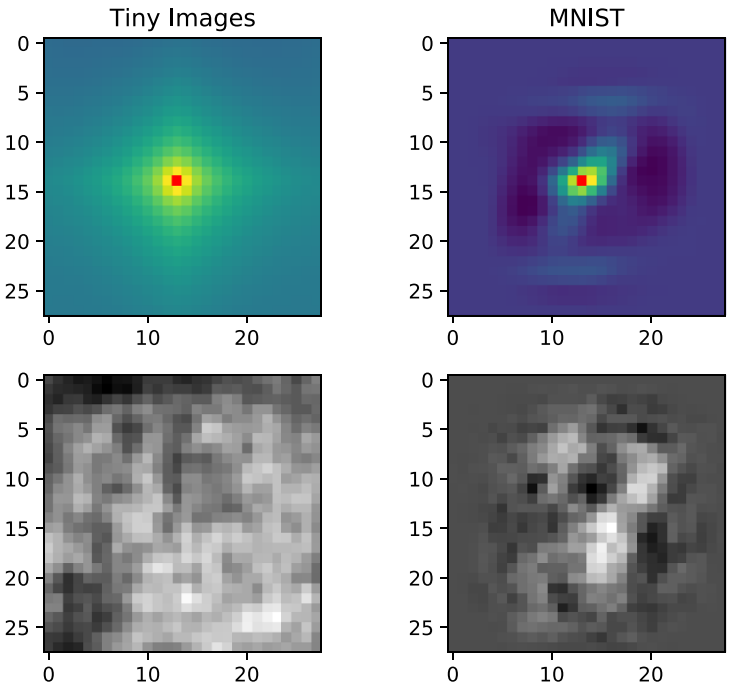


Figure 10. The covariances (top row) and sample images (bottom row) from GRMFs fit to the Tiny Images and MNIST datatsets. The covariance values are calculated with respect to the central pixel highlighted in red, with brighter colors indicating larger values. The Tiny Images covariance plot shows a strong nearest-neighbor pattern, while the MNIST plot has a more complicated and long-range structure. The sample images show some structure, but are not identifiable as a digit or object.

ORCID iDs

Ian Convy https://orcid.org/0000-0003-1818-2677
William Huggins https://orcid.org/0000-0003-2735-1380
Haoran Liao https://orcid.org/0000-0002-6399-006X
K Birgitta Whaley https://orcid.org/0000-0002-7164-4757

References

[1] Kolda T and Bader B 2009 Tensor decompositions and applications *SIAM Rev.* **51** 455–500
[2] Hackbusch W 2012 *Tensor Spaces and Numerical Tensor Calculus (Springer Series in Computational Mathematics)* (Berlin: Springer)
[3] Bridgeman J C and Chubb C T 2017 Hand-waving and interpretive dance: an introductory course on tensor networks *J. Phys. A: Math. Theor.* **50** 223001
[4] Eisert J 2013 Entanglement and tensor network states (arXiv:1308.3318 [cond-mat, physics: quant-ph])
[5] Verstraete F, Cirac J I and Murg V 2008 Matrix product states, projected entangled pair states and variational renormalization group methods for quantum spin systems *Adv. Phys.* **57** 143–224
[6] White S R 1992 Density matrix formulation for quantum renormalization groups *Phys. Rev. Lett.* **69** 2863–6
[7] Vidal G 2003 Efficient classical simulation of slightly entangled quantum computations *Phys. Rev. Lett.* **91** 147902
[8] Cohen N, Sharir O and Shashua A 2016 On the expressive power of deep learning: a tensor analysis *Conf. on Learning Theory* (PMLR) pp 698–728
[9] Stoudenmire E M and Schwab D J 2016 Supervised learning with quantum-inspired tensor networks (arXiv:1605.05775 [cond-mat, stat])
[10] Huggins W, Patil P, Mitchell B, Whaley K B and Stoudenmire E M 2019 Towards quantum machine learning with tensor networks *Quantum Sci. Technol.* **4** 024001
[11] Stoudenmire E M 2018 Learning relevant features of data with multi-scale tensor networks *Quantum Sci. Technol.* **3** 034003
[12] Cheng S, Wang L and Zhang P 2020 Supervised learning with projected entangled pair states (arXiv:2009.09932 [cond-mat, physics: quant-ph,stat])
[13] Schollwöck U 2011 The density-matrix renormalization group in the age of matrix product states *Ann. Phys., NY* **326** 96–192
[14] Eisert J, Cramer M and Plenio M B 2010 Area laws for the entanglement entropy—a review *Rev. Mod. Phys.* **82** 277–306
[15] Orus R 2014 A practical introduction to tensor networks: matrix product states and projected entangled pair states *Ann. Phys., NY* **349** 117–58
[16] Vidal G 2008 A class of quantum many-body states that can be efficiently simulated *Phys. Rev. Lett.* **101** 110501
[17] LeCun Y, Cortes C and Burges C MNIST handwritten digit database (available at: http://yann.lecun.com/exdb/mnist/)
[18] Torralba A, Fergus R and Freeman W 2008 80 million tiny images: a large data set for nonparametric object and scene recognition *IEEE Trans. Pattern Anal. Mach. Intell.* **30** 1958–70
[19] Oseledets I V 2011 Tensor-train decomposition *SIAM J. Sci. Comput.* **33** 2295–317
[20] Biamonte J and Bergholm V 2017 Tensor networks in a nutshell (arXiv:1708.00006 [cond-mat, physics: gr-qc,physics: hep-th,physics: math-ph,physics: quant-ph])
[21] Biamonte J 2020 Lectures on quantum tensor networks (arXiv:1912.10049 [cond-mat, physics: math-ph,physics: quant-ph])
[22] Seber G A F and Lee A J 2012 *Linear Regression Analysis* (New York: Wiley)
[23] Hastie T, Tibshirani R and Friedman J 2009 *The Elements of Statistical Learning: Data Mining, Inference and Prediction (Springer Series in Statistics)* 2nd edn (Berlin: Springer)
[24] Shi Y, Duan L and Vidal G 2006 Classical simulation of quantum many-body systems with a tree tensor network *Phys. Rev.* A **74** 022320
[25] Liu D, Ran S J, Wittek P, Peng C, García R B, Su G and Lewenstein M 2019 Machine learning by unitary tensor network of hierarchical tree structure *New J. Phys.* **21** 073059
[26] Reyes J A and Stoudenmire E M 2021 Multi-scale tensor network architecture for machine learning *Mach. Learn.: Sci. Technol.* **2** 035036
[27] Cong I, Choi S and Lukin M D 2019 Quantum convolutional neural networks *Nat. Phys.* **15** 1273–8
[28] Schrödinger E 1935 Discussion of probability relations between separated systems *Math. Proc. Camb. Phil. Soc.* **31** 555–63
[29] Plenio M B and Virmani S 2005 An introduction to entanglement measures (arXiv:0504163v3)
[30] Ekert A and Knight P L 1995 Entangled quantum systems and the Schmidt decomposition *Am. J. Phys.* **63** 415–23
[31] Hitchcock F L 1927 The expression of a tensor or a polyadic as a sum of products *J. Math. Phys.* **6** 164–89
[32] Evenbly G and Vidal G 2011 Tensor network states and geometry *J. Stat. Phys.* **145** 891–918
[33] Page D N 1993 Average entropy of a subsystem *Phys. Rev. Lett.* **71** 1291–4
[34] Hastings M B 2007 An area law for one-dimensional quantum systems *J. Stat. Mech.* **2007** 08024
[35] Cramer M, Eisert J, Plenio M B and Dreißig J 2006 Entanglement-area law for general bosonic harmonic lattice systems *Phys. Rev.* A **73** 012309
[36] Vidal G, Latorre J I, Rico E and Kitaev A 2003 Entanglement in quantum critical phenomena *Phys. Rev. Lett.* **90** 227902
[37] Ebrahimi N, Soofi E S and Soyer R 2010 Information measures in perspective: information measures in perspective *Int. Stat. Rev.* **78** 383–412
[38] Cover T M and Thomas J A 1991 *Elements of Information Theory* (New York: Wiley)
[39] Low G H, Yoder T J and Chuang I L 2014 Quantum inference on Bayesian networks *Phys. Rev.* A **89** 062315
[40] Wu S, Poulsen U V and Mølmer K 2009 Correlations in local measurements on a quantum state and complementarity as an explanation of nonclassicality *Phys. Rev.* A **80** 032319
[41] Cover T M and Thomas J A 1991 *Elements of Information Theory (Wiley Series in Telecommunications)* 6th edn (New York: Wiley)
[42] Martyn J, Vidal G, Roberts C and Leichenauer S 2020 Entanglement and tensor networks for supervised image classification (arXiv:2007.06082 [quant-ph, stat])
[43] Paninski L 2003 Estimation of entropy and mutual information *Neural Comput.* **15** 1191–253
[44] Moddemeijer R 1989 On estimation of entropy and mutual information of continuous distributions *Signal Process.* **16** 233–48
[45] Steuer R, Kurths J, Daub C O, Weise J and Selbig J 2002 The mutual information: detecting and evaluating dependencies between variables *Bioinformatics* **18** S231–40
[46] Epanechnikov V A 1969 Non-parametric estimation of a multivariate probability density *Theory Probab. Appl.* **14** 153–8
[47] Moon Y I, Rajagopalan B and Lall U 1995 Estimation of mutual information using kernel density estimators *Phys. Rev.* E **52** 2318–21
[48] Kraskov A, Stögbauer H and Grassberger P 2004 Estimating mutual information *Phys. Rev.* E **69** 066138
[49] Koeman M and Heskes T 2014 Mutual information estimation with random forests *Neural Information Processing Lecture Notes in Computer Science* ed C K Loo, K S Yap, K W Wong, A Teoh and K Huang (New York: Springer International Publishing) pp 524–31
[50] Belghazi M I, Baratin A, Rajeswar S, Ozair S, Bengio Y, Courville A and Hjelm R D 2018 Mine: mutual information neural estimation (arXiv:1801.04062 [cs, stat])

[51] Poole B, Ozair S, Oord A V D, Alemi A and Tucker G 2019 On variational bounds of mutual information *Int. Conf. on Machine Learning* (PMLR) pp 5171–80
[52] Ruderman A, Reid M D, García-García D and Petterson J 2012 Tighter variational representations of f-divergences via restriction to probability measures *Proc. 29th Int. Conf. on Int. Conf. on Machine Learning* pp 1155–62
[53] Ramos D, Franco-Pedroso J, Lozano-Diez A and Gonzalez-Rodriguez J 2018 Deconstructing cross-entropy for probabilistic binary classifiers *Entropy* **20** 208
[54] Rue H, Held L and Held L 2005 *Gaussian Markov Random Fields: Theory and Applications* (London: Chapman and Hall)
[55] Kanan C and Cottrell G W 2012 Color-to-grayscale: does the method matter in image recognition? *PLoS One* **7** e29740
[56] Pires C A L and Perdigão R A P 2012 Minimum mutual information and non-gaussianity through the maximum entropy method: theory and properties *Entropy* **14** 1103–26
[57] Li W, von Delft J and Xiang T 2012 Efficient simulation of infinite tree tensor network states on the Bethe lattice *Phys. Rev.* B **86** 195137
[58] Kong W, Vanderburg C R, Gunshin H, Rogers J T and Huang X 2008 A review of independent component analysis application to microarray gene expression data *BioTechniques* **45** 501–20
[59] Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I and Abbeel P 2016 InfoGAN: interpretable representation learning by information maximizing generative adversarial nets *Proc. 30th Int. Conf. on Neural Information Processing Systems NIPS'16* (Curran Associates Inc.) pp 2180–8
[60] Cheng S, Chen J and Wang L 2018 Information perspective to probabilistic modeling: Boltzmann machines versus born machines *Entropy* **20** 583
[61] McAllester D and Stratos K 2020 Formal limitations on the measurement of mutual information *Int. Conf. on Artificial Intelligence and Statistics* (PMLR) pp 875–84