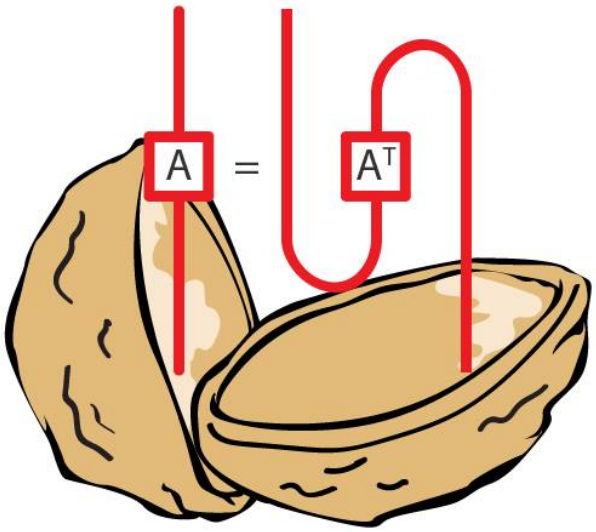
Quantum Tensor Networks in a Nutshell

Jacob Biamonte[1,2,*] and Ville Bergholm[1,†]

[1] *Quantum Software Initiative*
Skolkovo Institute of Science and Technology,
Skoltech Building 3, Moscow 143026, Russia
[2] *Institute for Quantum Computing*
University of Waterloo, Waterloo, N2L 3G1 Ontario, Canada

Tensor network methods are taking a central role in modern quantum physics and beyond. They can provide an efficient approximation to certain classes of quantum states, and the associated graphical language makes it easy to describe and pictorially reason about quantum circuits, channels, protocols, open systems and more. Our goal is to explain tensor networks and some associated methods as quickly and as painlessly as possible. Beginning with the key definitions, the graphical tensor network language is presented through examples. We then provide an introduction to matrix product states. We conclude the tutorial with tensor contractions evaluating combinatorial counting problems. The first one counts the number of solutions for Boolean formulae, whereas the second is Penrose's tensor contraction algorithm, returning the number of 3-edge-colorings of 3-regular planar graphs.



* jacob.biamonte@qubit.org; www.QuamPlexity.org
† ville.bergholm@iki.fi

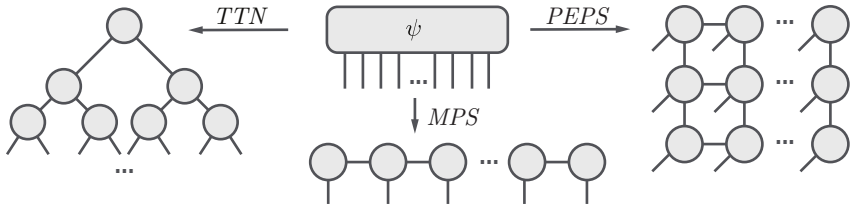
1. QUANTUM LEGOS

Tensors are a mathematical concept that encapsulates and generalizes the idea of multilinear maps, i.e. functions of multiple parameters that are linear with respect to every parameter. A tensor network is simply a countable collection of tensors connected by contractions. 'Tensor network methods' is the term given to the entire collection of associated tools, which are regularly employed in modern quantum information science, condensed matter physics, mathematics and computer science.

Tensor networks come with an intuitive graphical language that can be used to reason about them. This diagrammatic language dates back to at least the early 1970s by Roger Penrose [1]. These methods have seen many advancements and adaptations to different domains of physics, mathematics and computer science. An important milestone was David Deutsch's use of the diagrammatic notation in quantum computing, developing the *quantum circuit* (a.k.a. quantum computational network) model [2]. Quantum circuits are a special class of tensor networks, in which the arrangement of the tensors and their types are restricted. A related diagrammatic language slightly before that is due to Richard Feynman [3]. The quantum circuit model—now well over two decades old—is widely used to describe quantum algorithms and their experimental implementations, to quantify the resources they use (by e.g. counting the quantum gates required), to classify the entangling properties and computational power of specific gate families, and more.

There is now a lot of excitement about tensor network algorithms—for reviews see [4–16]. Some of the best known applications of tensor networks are 1D Matrix Product States (MPS), Tensor Trains (TT), Tree Tensor Networks (TTN), the Multi-scale Entanglement Renormalization Ansatz (MERA), Projected Entangled Pair States (PEPS)—which generalize matrix product states to higher dimensions—and various other renormalization methods [5–8, 12, 15, 17]. The excitement is based on the fact that certain classes of quantum systems can now be simulated more efficiently, studied in greater detail, and this has opened new avenues for a greater understanding of certain physical systems.

These methods approximate a complicated quantum state using a tensor network with a simplistic, regular structure—essentially applying lossy data compression that preserves the most important properties of the quantum state. To give the reader a rough idea how these methods work, below we conceptually depict how the quantum state ψ could be represented (or approximated) using tensor networks in various ways.



(1)

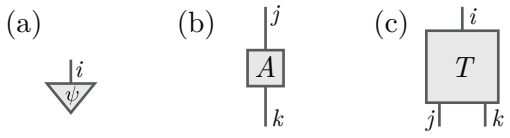
We assume that most readers will have a basic understanding of some quantum theory, linear algebra and tensors. In Appendix A, we provide a short mathematical definition of tensors and tensor products. However, readers may wish to skip these definitions and for now proceed with a more informal or intuitive understanding of the idea.

There are several notational approaches to tensors. We begin by using abstract index notation, and then explain how it connects to the Dirac notation used in quantum computing. The connection between tensor networks and quantum circuits is elucidated at the end of Section 2.

CONTENTS

2. FROM TENSORS TO NETWORKS

1. **Drawing tensors.** In the tensor diagram notation, a tensor is a labelled shape such as a box, oval or triangle, with zero or more open output legs (or *arms*) pointing up, and zero or more open input legs pointing down. Individual arms and legs each correspond to upper and lower indices, respectively.[1] The arm and leg wires may be labelled with the indices they represent, or with the vector spaces they correspond to, if necessary. An order-$(0,0)$ tensor without any open arms or legs is simply a complex number.



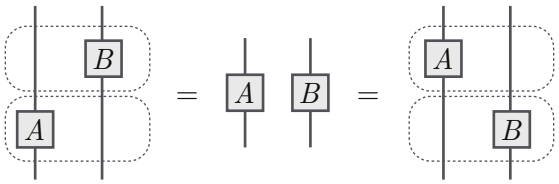
(2)

For example, diagram (a) above represents the tensor ψ^i with a single upper index (a vector), diagram (b) the tensor $A^j{}_k$ (a matrix), and diagram (c) the tensor $T^i{}_{jk}$.

2. **Tensor juxtaposition.** When two or more disconnected tensors appear in the same diagram they are multiplied together using the tensor product. In quantum physics notation, they would have a tensor product sign $\otimes$ between them. In the abstract index notation the tensor product sign is omitted.

[1] This conformity is simplest to get started. However, to conserve space, tensor diagrams are often rotated without any warning 90 degrees clockwise. In practice this should be clear from the context.

Tensors can be freely moved past each other. This is sometimes called planar deformation.

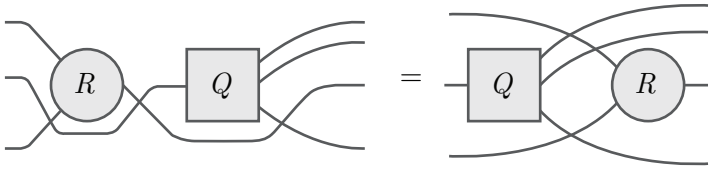


(3)

From the diagram above, using equations we have

$$(\mathbb{1} \otimes B)(A \otimes \mathbb{1}) = A \otimes B = (A \otimes \mathbb{1})(\mathbb{1} \otimes B), \tag{4}$$

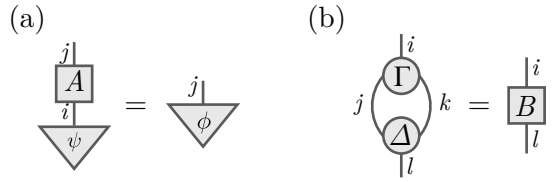
where we make use of the wire also playing the role of the identity tensor $\mathbb{1}$—detailed in Section 3. As we shall soon see, wires are allowed to cross tensor symbols and other wires, as long as the wire endpoints are not changed. This is one reason why tensor diagrams are often simpler to deal with than their algebraic counterparts.



(5)

In the diagram above we did not label the wires, since it is an arbitrary assignment. If we did, we could for example denote it as $Q^{deg}{}_{b} R^{f}{}_{ac}$.

3. **Connecting wires.** Connecting two tensor legs with a wire means that the corresponding indices are contracted (summed over).



(6)

In diagram (a) above we find a matrix multiplying a vector which results in another vector. Diagram (a) is equivalent to the expression

$$A^{j}{}_{i} \psi^{i} = \phi^{j}, \tag{7}$$

where we notice that the wire labeled i is fully connected, and hence the corresponding index is summed over. We used the *Einstein summation convention*, in which any index that appears exactly twice in a term is summed over.

In (b) we face a slightly more complicated case, departing from familiar vectors and matrices, contracting two indices between two order-3 tensors. Diagram (b) is equivalent to

$$\Gamma^{i}{}_{jk} \Delta^{jk}{}_{l} = B^{i}{}_{l}. \tag{8}$$

Together, two or more tensors in a diagram form a *tensor network*. If none of the tensors have any open arms or legs the network is said to be fully contracted: it evaluates to some complex number, a scalar.

You have probably noticed that we are being strict and are leaving spaces to identify the order of indices as we read them left to right across the page (e.g. in $\Gamma^i{}_{jk}$ we have i, followed by jk). In Section 3 we are going to talk about this in more detail, and justify why it is not necessary: we are doing it here for ease of illustration.

Some authors like to put arrows on the tensor diagram wires [18], to denote the difference between vector spaces and their duals.[2] In this work we mostly deal with finite-dimensional vector spaces over real or complex numbers, and for each vector space V we pick a preferred basis (called the *computational basis* in quantum computing). The basis can be used to define an inner product, turn V into a Hilbert space, and establish a bijective mapping between V and its dual V^*. We then use this mapping to equate dual vectors in V^* with their counterparts in V. In this case arrows on the wires add nothing essential, so we omit them.

4. **Connection to quantum computing notation.** As mentioned, tensors are multilinear maps. They can be expanded in any given basis, and expressed in terms of their components. In quantum information science one often introduces a *computational basis* $\{|k\rangle\}_k$ for each Hilbert space and expands the tensors in it, using kets ($|\ \rangle$) for vectors and bras ($\langle\ |$) for dual vectors:

$$T = \sum_{ijk} T^i{}_{jk}\, |i\rangle\langle jk|. \tag{9}$$

Here $T^i{}_{jk}$ is understood not as abstract index notation but as the actual components of the tensor in the computational basis. In practice there is little room for confusion. The Einstein summation convention is rarely used in quantum information science, hence we write the sum sign explicitly.

So far we have explained how tensors are represented in tensor diagrams, and what happens when wires are connected. The ideas are concluded by four examples; we urge the reader to work through the examples and check the results for themselves.

The first example introduces a familiar structure from linear algebra in tensor form. The next two examples come from quantum entanglement theory—see connecting tensor networks with invariants [19, 20]. The fourth one showcases quantum circuits, a subclass of tensor networks widely used in the field of quantum information. The examples are chosen to illustrate properties of tensor networks and should be self-contained.

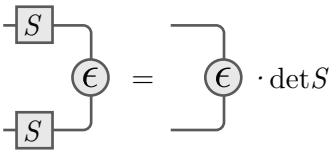
Example 1 (The ϵ tensor)**.** A tensor is said to be fully antisymmetric if swapping any pair of indices will change its sign: $A_{ij} = -A_{ji}$. The ϵ tensor is used to represent the fully antisymmetric Levi-Civita symbol, which in two dimensions can be expressed as

$$\epsilon_{00} = \epsilon_{11} = 0, \qquad \epsilon_{01} = -\epsilon_{10} = 1. \tag{10}$$

The ϵ tensor can be used to compute the determinant of a matrix. In two dimensions we have

$$\det(S) = \epsilon_{ij} S^i{}_0 S^j{}_1. \tag{11}$$

Using this we obtain



(12)

[2] See Appendix A for the definition of a dual space.

as can be seen by labeling the wires in the diagram. In equational form this is

$$\epsilon_{ij} S^{i}{}_{m} S^{j}{}_{n} = \det(S)\, \epsilon_{mn}. \tag{13}$$

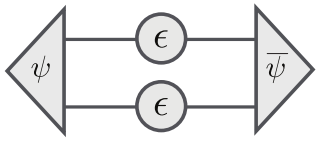
In terms of quantum mechanics, ϵ corresponds to the two-qubit singlet state:

$$\frac{1}{\sqrt{2}}|\epsilon\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \tag{14}$$

This quantum state is invariant under any transformation of the form $U \otimes U$, where U is a 2×2 unitary, as it only gains an unphysical global phase factor $\det(U)$.

Example 2 (Concurrence and entanglement)**.** Given a two-qubit pure quantum state $|\psi\rangle$, its *concurrence* $C(\psi) = |C'(\psi)|$ is the absolute value of the following tensor network expression [21]:

$C'(\psi) =$



(15)

Here $\overline{\psi}$ is the complex conjugate of ψ in the computational basis. The concurrence is an entanglement monotone, a function from states to nonnegative real numbers that measures how entangled the state is. $|\psi\rangle$ is entangled if and only if the concurrence is greater than zero.

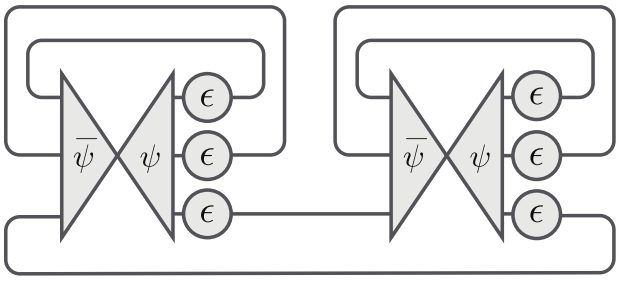
Consider now what happens when we act on $|\psi\rangle$ by an arbitrary local unitary operation, i.e. $|\psi'\rangle = (U_1 \otimes U_2)|\psi\rangle$. Using the result of Example 1 we obtain

$$C\left((U_1 \otimes U_2)|\psi\rangle\right) = C(\psi)|\det(U_1)\det(U_2)|. \tag{16}$$

Due to the unitarity $|\det U_1| = |\det U_2| = 1$, which means that the value of the concurrence is *invariant* (i.e. does not change) under local unitary transformations. This is to be expected, as local unitaries cannot change the amount of entanglement in a quantum state. We will revisit concurrence in Example 11.

More complicated invariants can also be expressed as tensor networks [19]. We will leave it to the reader to write the following network as an algebraic expression:

$\tau'(\psi) =$



(17)

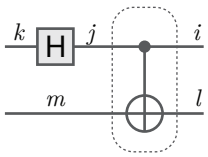
If $|\psi\rangle$ is a 3-qubit quantum state, $\tau(\psi) = 2|\tau'(\psi)|$ represents the entanglement invariant known as the 3-tangle [22]. It is possible to form invariants also without using the epsilon tensor. For example, the following expression represents the 3-qubit entanglement invariant known as the Kempe invariant [23]:

$$K(\psi) = \psi^{ijk}\, \overline{\psi}_{ilm}\, \psi^{nlo}\, \overline{\psi}_{pjo}\, \psi^{pqm}\, \overline{\psi}_{nqk}. \tag{18}$$

The studious reader would draw the equivalent tensor network.

Example 3 (Quantum circuits)**.** Quantum circuits are a restricted subclass of tensor networks that is widely used in the field of quantum information. In a quantum circuit diagram each horizontal wire represents the Hilbert space associated with a quantum subsystem, typically a single qubit. The tensors attached to the wires represent unitary propagators acting on those subsystems, and are called *quantum gates*. Additional symbols may be used to denote measurements. The standard notation is described in [24].

Here we will consider a simple quantum circuit that can generate entangled Bell states. It consists of two tensors, a Hadamard gate (H) and a controlled NOT gate (CNOT, denoted by the symbol inside the dashed region):



(19)

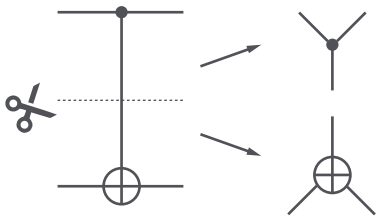
The CNOT and Hadamard gates are defined as

$$\mathsf{CNOT} = \sum_{ab} |a, a \oplus b\rangle\langle a, b| \quad \text{and} \tag{20}$$

$$\mathsf{H} = \frac{1}{\sqrt{2}} \sum_{ab} (-1)^{ab} |a\rangle\langle b|, \tag{21}$$

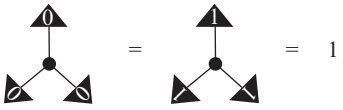
where the addition in the CNOT is modulo 2.[3] The reader should verify that acting on the quantum state $|00\rangle$ the above circuit yields the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, and acting on $|11\rangle$ it yields the singlet state $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$.

Example 4 (COPY and XOR tensors)**.** One can view the CNOT gate itself as a contraction of two order-three tensors [25]:



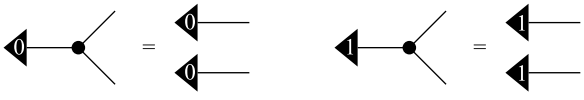
(22)

The top tensor (• with three legs) is called the COPY tensor. It equals unity when all the indices are assigned the same value (0 or 1), and vanishes otherwise:



(23)

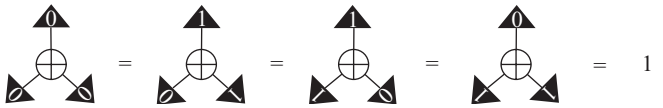
Hence, COPY acts to copy the binary inputs 0 and 1:



(24)

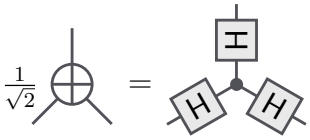
[3] Addition modulo 2: $1 \oplus 1 = 0 \oplus 0 = 0, \quad 1 \oplus 0 = 0 \oplus 1 = 1$.

The bottom tensor ($\oplus$ with three legs) is called the parity or XOR tensor. It equals unity when the index assignment contains an even number of 1s, and vanishes otherwise:



(25)

The XOR and COPY tensors are related via the Hadamard gate as



(26)

Thus one can think of XOR as being a (scaled) copy operation in another basis:

$$\frac{1}{\sqrt{2}}\mathsf{XOR}|+\rangle = |+\rangle|+\rangle, \tag{27a}$$

$$\frac{1}{\sqrt{2}}\mathsf{XOR}|-\rangle = |-\rangle|-\rangle, \tag{27b}$$

where $|+\rangle := \mathsf{H}|0\rangle$ and $|-\rangle := \mathsf{H}|1\rangle$. In terms of components,

$$\mathsf{COPY}^{ij}{}_{k} = (1-i)(1-j)(1-k) + ijk, \tag{28a}$$

$$\mathsf{XOR}^{qr}{}_{s} = 1 - (q+r+s) + 2(qr+qs+sr) - 4qrs. \tag{28b}$$

The CNOT gate is now obtained as the tensor contraction

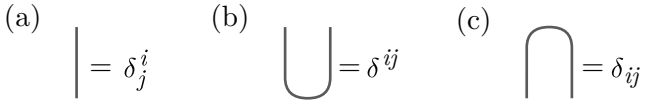
$$\sum_{m} \mathsf{COPY}^{qm}{}_{i}\, \mathsf{XOR}^{r}{}_{mj} = \mathsf{CNOT}^{qr}{}_{ij}. \tag{29}$$

The COPY and XOR tensors will be explored further in later examples and have many convenient properties [26, 27, 39].

3. BENDING AND CROSSING WIRES

"*It now ceases to be important to maintain a distinction between upper and lower indices.*"
– Roger Penrose, 1971 [1]

1. **Cups and caps.** As explained in the previous section, wires are used to denote the contraction of pairs of tensor indices. However, it is often useful to interpret certain wire structures as independent tensors of their own. We start with three of these special *wire tensors* that allow one to rearrange the arms and legs of another tensor:



(30)

The identity tensor (a) is used for index contraction by connecting the corresponding legs. The cup (b) and the cap (c) raise and lower tensor indices by bending the corresponding tensor legs.[4] Expanding them in the computational basis we obtain

$$\mathbb{1} = \sum_{ij} \delta^i{}_j |i\rangle\langle j| = \sum_k |k\rangle\langle k|, \tag{31}$$

$$|\cup\rangle = \sum_{ij} \delta^{ij} |ij\rangle = \sum_k |kk\rangle, \tag{32}$$

$$\langle\cap| = \sum_{ij} \delta_{ij} \langle ij| = \sum_k \langle kk|. \tag{33}$$

In a quantum information context, the cup also corresponds to an (unnormalized) Bell state, generalized so that it is not defined just for qubits.

2. **Snake equation.** One can raise and then lower an index or vice versa, which amounts to doing nothing at all. This idea is captured diagrammatically by the so called *snake* or *zig-zag equation* [1].

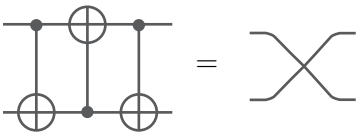
(34)

In abstract index notation it is expressed succinctly as $\delta^{ij}\delta_{jk} = \delta^i{}_k = \delta_{kj}\delta^{ji}$.

3. **SWAP gate.** Crossing two wires (as in diagram (a) below) can be thought of as swapping the relative order of two vector spaces. It corresponds to the SWAP gate used in quantum computing. If both wires represent the same vector space, it can be alternatively understood as swapping the states of the two subsystems.

(35)

Equation (b) illustrates that the SWAP operation is self inverse. It may be written as $\mathsf{SWAP}^{ij}{}_{kl} = \delta^j{}_k \delta^i{}_l$, or expanded in the computational basis as $\mathsf{SWAP} = \sum_{ij} |ij\rangle\langle ji|$. It also has a well-known implementation in terms of three CNOT gates as

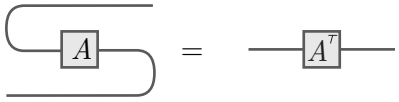


(36)

SWAP is the simplest nontrivial example of a permutation tensor. More complicated permutations may be built out of the $\delta^i{}_j$ tensors in an obvious way. We will return to this idea in Example 7.

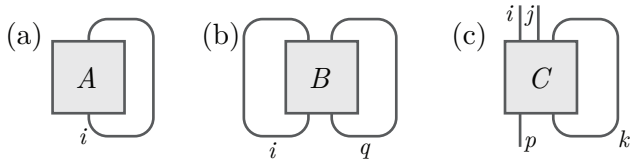
[4] Some readers familiar with relativity will note similarities with the metric tensor—here we will always work in a flat Euclidean space, meaning the metric tensors are trivial.

4. **Transpose.** Given $A^i{}_j$, we may reverse the positions of its indices using a cup and a cap. This is equivalent to transposing the corresponding linear map in the computational basis:



(37)

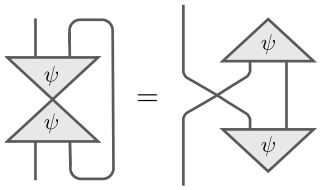
5. **Trace.** In the tensor diagram notation, trace is given by appropriately joining all the output wires of a tensor to corresponding input wires. Diagram (a) below represents the trace $A^i{}_i$. Diagram (b) represents the trace $B^{iq}{}_{iq}$.



(38)

Partial trace means contracting only some of the outputs with their corresponding inputs, such as with the tensor $C^{ijk}{}_{pk}$ shown in diagram (c).

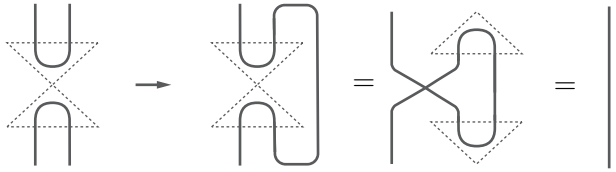
Example 5 (Partial trace)**.** The following is an early rewrite representing entangled pairs due to Penrose [28].



(39)

The diagram on the left represents the partial trace of $|\psi\rangle\langle\psi|$ over the second subsystem. Readers can prove that this equality follows by interpreting the bent wires as cups and caps, and the crossing wires as SWAPs.

Example 6 (Partial trace of Bell states)**.** Continuing on from Example 5, if we choose $|\psi\rangle = |\cup\rangle$, i.e. $|\psi\rangle$ is an unnormalized Bell state, we obtain



(40)

Example 7 (Relation between ϵ and SWAP)**.** For any order-2 tensor T^{ij} we can define its *antisymmetrization* as $T^{[ij]} = \frac{1}{2}(T^{ij} - T^{ji})$. Here we used the notation of putting brackets around a group of indices—$[ij]$—to denote their antisymmetrization. Only indices of the same dimension may be antisymmetrized (otherwise the expression would be undefined for some index values).

The fully antisymmetric ϵ tensor from Example 1 has an interesting relation to the SWAP gate:

(41)

or alternatively

$$\epsilon^{kl}\epsilon_{ij} = \delta^k{}_i\delta^l{}_j - \delta^k{}_j\delta^l{}_i. \tag{42}$$

It is now easy to show that for any tensor T^{ij} (for which both indices are two-dimensional) we can write $T^{[kl]} = \frac{1}{2}\epsilon^{kl}\epsilon_{ij}T^{ij}$.

Both the concept of antisymmetrization and the epsilon tensor can be extended to more than two indices. The *antisymmetrizer* of n d-dimensional vector spaces is an order-(n,n) tensor $A^{i_1\cdots i_n}{}_{j_1\cdots j_n}$. It can be expressed as the sum of all n-element permutations multiplied by their signatures.[5] It antisymmetrizes n d-dimensional indices by contraction:

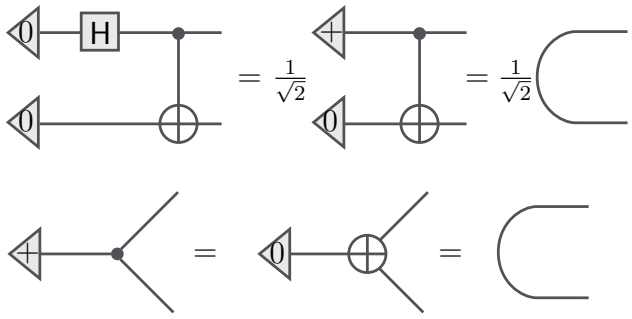
$$T^{[i_1\cdots i_n]} = A^{i_1\cdots i_n}{}_{j_1\cdots j_n}T^{j_1\cdots j_n}. \tag{43}$$

When $d < n$ the only possible antisymmetric combination is a zero tensor, and the corresponding A vanishes identically.

For the general order-$(0,n)$ epsilon tensor, all the n vector spaces need to be n-dimensional. We then define $\epsilon_{012\ldots(n-1)} = 1$, and all the other components are fixed by requiring complete antisymmetry, i.e. change of sign under the interchange of any two indices. In particular, if any index value is repeated the corresponding component is zero. Now $\frac{1}{n!}\epsilon^{i_1\cdots i_n}\epsilon_{j_1\cdots j_n}$ is an antisymmetrizer.

We shall use order-three epsilon tensors to count graph edge colorings by tensor contraction in Section 6 2.

Example 8 (Quantum circuits for cups and epsilon states)**.** The quantum circuit from Example 3 is typically used to generate entangled qubit pairs. For instance, acting on the state $|00\rangle$ yields the familiar Bell state—as a tensor network, this is equal to a normalized cup. Here we also show the mathematical relationship the XOR and COPY tensors have with the cup (here $|+\rangle := |0\rangle + |1\rangle$):

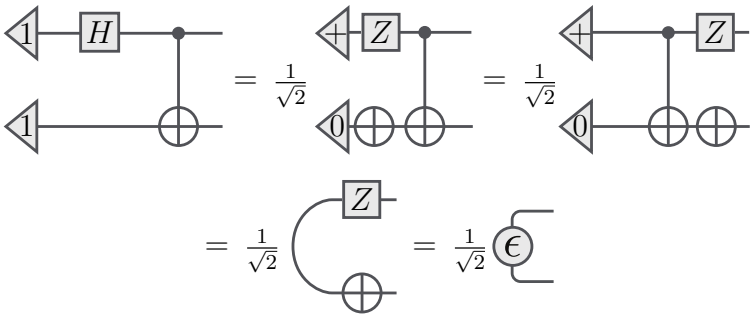


(44)

Similarly, one can use the circuit (19) to generate the epsilon state. Let us denote the Pauli matrices by $X := |0\rangle\langle 1| + |1\rangle\langle 0|$, $Y := -i|0\rangle\langle 1| + i|1\rangle\langle 0|$ and $Z := |0\rangle\langle 0| - |1\rangle\langle 1|$. The

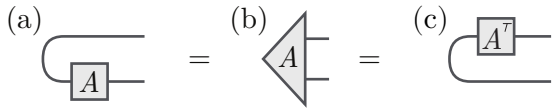
[5] The signature of a permutation σ is $(-1)^{N(\sigma)}$, where $N(\sigma)$ is the number of pairwise swaps it contains.

Z gate commutes with the COPY tensor, and the X or NOT gate (which we denote with $\oplus$) commutes with XOR. Commuting those tensors to the right hand side, allows us to apply Eq. (44). Making use of the Pauli algebra identity $ZX = iY$, one recovers the epsilon state:



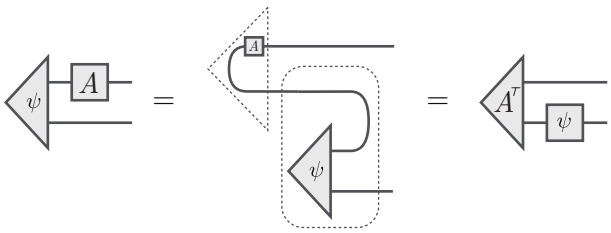
(45)

Example 9 (Map-state duality)**.** The index raising and lowering using cups and caps can be interpreted as a linear map between bipartite vectors and linear maps leading to a relationship between quantum states and operators acting on them. We will start with the linear map A. Raising the second index using a cup (a) yields a tensor with two output legs (b), i.e. something that can be interpreted as a bipartite vector $|A\rangle$.



(46)

Finally, using the snake equation (34) together with the equality (37) we can see that this is equal to (c) the transposed map $A^\top$ with a cup raising the input index to the other side. Indeed, a tensor may be moved around a cup or a cap by transposing it. The relationship (46) arises in practice in the following scenario from quantum information science:



(47)

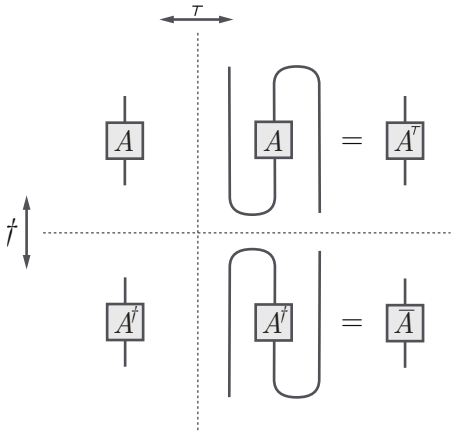
Here an entangled state $|\psi\rangle$ acted on by a map A can instead be viewed as a map ψ acting on a state $|A^\top\rangle$.[6] This is a diagrammatic form of map-state duality underlying bipartite entanglement evolution [29, 30]. See e.g. the survey [15] which includes a detailed discussion on reshaping tensors.

6. **Dagger and complex conjugation.** If the order-(p, q) tensor T is a map between Hilbert spaces, we may define its (Hermitian) adjoint $T^\dagger$, an order-(q, p) tensor, using the inner product: $\langle x, Ty\rangle = \langle T^\dagger x, y\rangle$ for all x, y. Diagrammatically the adjoint is obtained by mirroring the tensor network such that input and output wires switch places, the relative order of the tensors in the diagram is reversed, and each tensor symbol is decorated with a dagger (with $T^{\dagger\dagger} = T$).

[6] Note that the transpose in $|A^\top\rangle$ is purely to adhere to the convention in Eq. (46). In terms of numerics, the vector $|A\rangle$ is defined here as the matrix A vectorized columnwise. Consequently $|A^\top\rangle$ is the same matrix vectorized row-wise.

Similarly, the dagger operation maps Hilbert space ket vectors one-to-one to their dual bra vectors (and vice versa) in the sense of the Riesz representation theorem. This is denoted $|a\rangle^\dagger = \langle a|$. Note that we have been using this notation implicitly, as it should be familiar to many readers from basic quantum mechanics.

The dagger is an antilinear (or conjugate-linear) operation, $(cT + dU)^\dagger = \overline{c}\,T^\dagger + \overline{d}\,U^\dagger$, since we have to take the complex conjugate of the scalars c and d. This means that it cannot be represented by the linear cups and caps alone, unlike the transpose. However, it can be represented as transpose together with complex conjugation in the same basis. This is summarized in the following *adjoint square*.



(48)

Note that some authors mark one of the corners of each tensor box, and use the convention that mirroring a diagram across the horizontal plane corresponds to the † operation. We do not adhere to this convention and instead place the dagger on the symbol such as $A \to A^\dagger$.

For kets and bras, bending a wire on a ket yields the complex conjugate of the corresponding bra, and vice versa.

(49)

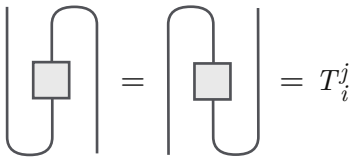
7. **Index position.** Now we will consider the set of operations formed by bending tensor wires forwards and backwards using cups and caps, as well as exchanging the order of wires using SWAP. If one conceptualizes a tensor as an array of numbers, these transforms correspond to array reshapes and reorderings. As the snake equation (34) shows, the action of cups and caps can be inverted, and SWAP is self inverse (35). This means that all possible configurations of a tensor's wires obtained using these operations are isomorphic.

As an example, given a tensor $T^i{}_j$ one can use cups and caps to naively rearrange the index elevations and positions, arriving at

$$T^i{}_j,\ T^{ij},\ T_{ij},\ T_i{}^j,\ T^j{}_i,\ T^{ji},\ T_{ji} \text{ and } T_j{}^i, \tag{50}$$

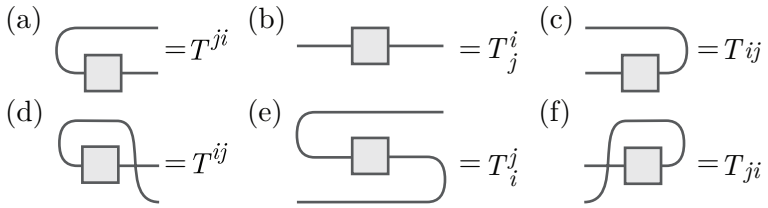
for a total of eight possible reshapes. If the tensor has more than two indices, one can additionally use SWAPs to arrange the indices in any relative order. Thus one might think that for a general n-index tensor there are $n! \cdot 2^n$ different ways of arranging the indices ($n!$ different permutations of the indices, with each index being either up or down). However, this way one overcounts the number of index configurations that are truly different. In our

example, in fact $T^i{}_j = T_j{}^i$ and $T_i{}^j = T^j{}_i$, as can be seen from the diagram below:



(51)

Consequently, the tensor T^i_j in actuality only has six unique reshapes:



(52)

More generally, one finds that the number of unique reshapes generated by cups, caps and SWAPs for an order-(p, q) tensor $T^{i_1 \cdots i_p}{}_{j_1 \cdots j_q}$ is $(p + q + 1)!$.

4. DIAGRAMMATIC SVD

In this section, we explain the diagrammatic version of the singular value decomposition (SVD). This method is at the heart of many numerical simulation algorithms in wide use today—we are explaining it as a prelude leading to Section 5. The SVD factors an arbitrary order-$(1, 1)$ tensor into well defined building blocks with simple properties: (i) an order-$(1, 1)$ diagonal tensor storing the singular values, and (ii) two order-$(1, 1)$ unitary tensors. As several tensor legs can always be grouped together to form a single leg, the method works for any tensor of order two or higher.

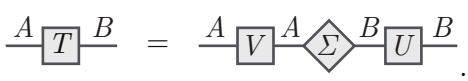
Interpreting order-$(1, 1)$ tensors as linear maps (or simply as matrices), we may use the SVD to factor any tensor $T : A \to B$ (for vector spaces A and B) as

$$T^b{}_a = U^b{}_j \Sigma^j{}_i V^i{}_a \,, \tag{53}$$

where U and V are unitary, and Σ is real, non-negative, and diagonal in the computational basis. Σ has the singular values $\{\sigma_k\}_k$ of T on its diagonal, typically arranged in a nonincreasing order: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min(\dim A, \dim B)} \geq 0$. It can be expanded as

$$\Sigma = \sum_k \sigma_k |k\rangle_B \langle k|_A. \tag{54}$$

Diagrammatically this is represented as



(55)

As will be seen in the next section, the SVD is centrally employed in the efficient representation of certain quantum states by tensor networks. The idea is to represent a small but physically relevant portion of the Hilbert space—such as low entanglement states—by repeated application of the SVD paired with low-rank approximations.

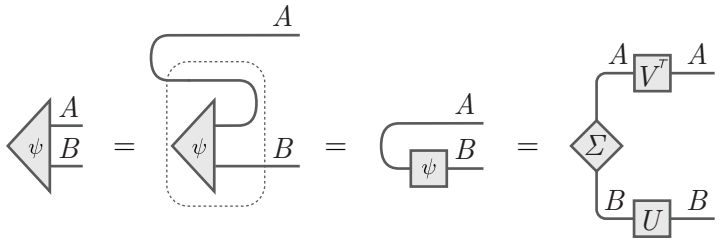
The rank of a matrix T is the number of non-zero singular values it has. To determine its optimal rank-r approximation (with $r < \text{rank}(T)$), we can turn to a classic theorem by Eckart and Young which was generalized by Mirsky. Given the SVD $T = U\Sigma V^\dagger$, we will discard $\text{rank}(T) - r$ smallest singular values in Σ by setting them to zero, obtaining Σ'. This process is often called trimming.

This gives rise to $T' = U\Sigma' V^\dagger$, an approximation of T. The Eckart-Young-Mirsky theorem states that

$$\|T - T'\| = \min_{\text{rank}(\hat{T}) \leq r} \|T - \hat{T}\| \tag{56}$$

for any unitarily invariant matrix norm.[7] Here $\hat{T}$ is any approximation to T of the same or lesser rank as T'. This implies that truncating or trimming Σ in this way yields as good of an approximation as one can expect. In the following section, we will specifically consider the induced error for such an approximation.

Using the wire bending techniques from Section 3, we immediately obtain the Schmidt decomposition as a corollary to the SVD:



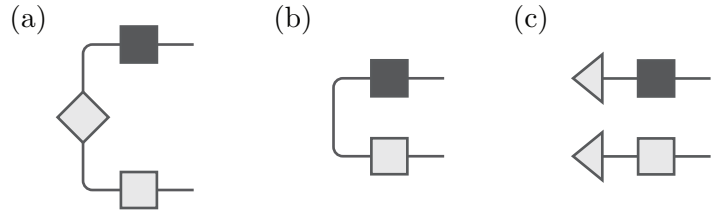
(57)

Given a vector $|\psi\rangle \in A \otimes B$ (for example a ket vector describing a pure state of a bipartite quantum system), we may use the snake equation to convert it into a linear map $\psi : A \to B$ (inside the dashed region). Now we apply the SVD on ψ as above. Diagram reorganization leads to the diagrammatic Schmidt decomposition

$$|\psi\rangle_{A\otimes B} = \sum_i \sigma_i |\varphi_i\rangle_A |\phi_i\rangle_B. \tag{58}$$

The singular values $\{\sigma_k\}_k$ now correspond to the Schmidt coefficients. If $|\psi\rangle$ is normalized, we have $\sum_k \sigma_k^2 = 1$.

Example 10 (Entanglement topology)**.** The topology of the bipartite quantum state $|\psi\rangle$ depends solely on the Schmidt coefficients $\{\sigma_k\}_k$.

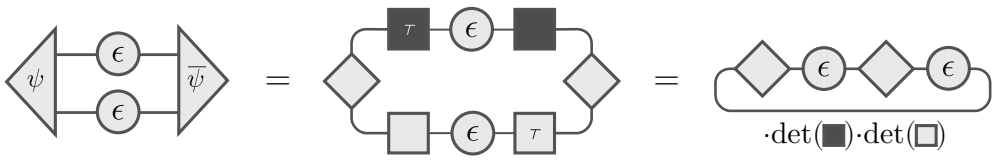


(59)

[7] Interested readers can try to get their hands on copies of C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," Psychometrika 1, (1936) and L. Mirsky, "Symmetric gauge functions and unitarily invariant norms," The Quarterly Journal of Mathematics 11:1, 50–59 (1960).

If the coefficients are all equal, we may replace the diamond in (a) with a cup tensor times a scaling factor as in diagram (b). This corresponds to a maximally entangled state. In the other extreme, illustrated in diagram (c), we have just one nonzero Schmidt coefficient $\sigma_1 = 1$. In this case the diagram breaks into two pieces and thus corresponds to a factorizable state. The number of nonzero Schmidt coefficients is called the *Schmidt rank* of the decomposition—see Def. 17 for the relation with Rényi entropy of order zero.

Example 11 (Concurrence—part II)**.** Continuing on from Example 2, one can apply the Schmidt decomposition to the tensor network defining the concurrence of a two-qubit state $|\psi\rangle$. We obtain

$C'(\psi) =$ 

(60)

and thus

$$C(\psi) = |C'(\psi)| = \left|\mathrm{Tr}((\epsilon\Sigma)^2)\right| = 2\sigma_1\sigma_2 = 2\,|\det(\psi)|. \tag{61}$$

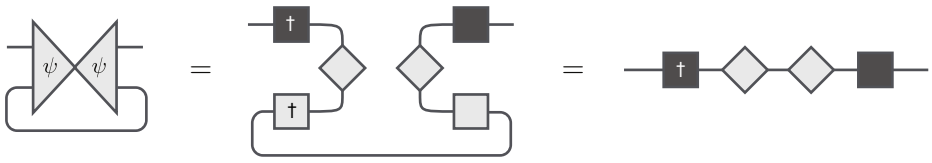
The unitaries ■ and □ vanish from the expression since $|\det(■)| = |\det(□)| = 1$. Given the basis expansion of the state,

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle, \tag{62}$$

its two Schmidt coefficients are given as

$$\sigma_k^2 = \frac{1}{2}\left(1 + (-1)^{k+1}\sqrt{1 - 4|ad - bc|^2}\right). \tag{63}$$

Example 12 (Purification backwards)**.** For any bipartite ket vector $|\psi\rangle$, the partial trace of $|\psi\rangle\langle\psi|$ yields a positive semidefinite operator. We can see this by using the Schmidt decomposition, and then reorganizing the diagram so that two of the unitaries cancel:



(64)

On the right hand side we have an eigendecomposition of a square matrix with strictly nonnegative eigenvalues $\{\sigma_k^2\}_k$, which can be interpreted as a density matrix if $|\psi\rangle$ is normalized. Conversely, any density matrix representing a mixed quantum state can be *purified*, or expressed as the partial trace of a bipartite pure state.

5. MATRIX PRODUCT STATES

Matrix product states (MPSs) are quantum states presented as a linear chain or ring of tensors. Any quantum state can be exactly represented in this form and the representation is known to approximate a class of 1D gapped systems efficiently [31]. We will explain the basic ideas of the MPS representation here and point the reader to [6, 7] and the references therein for additional information.

Given an n-party quantum state $|\psi\rangle$, fully describing this state generally requires an amount of information (or computer memory) that grows exponentially with n. If $|\psi\rangle$ represents the state of n qubits,

$$|\psi\rangle = \sum_{ij\cdots k} \psi_{ij\cdots k} |ij\cdots k\rangle, \tag{65}$$

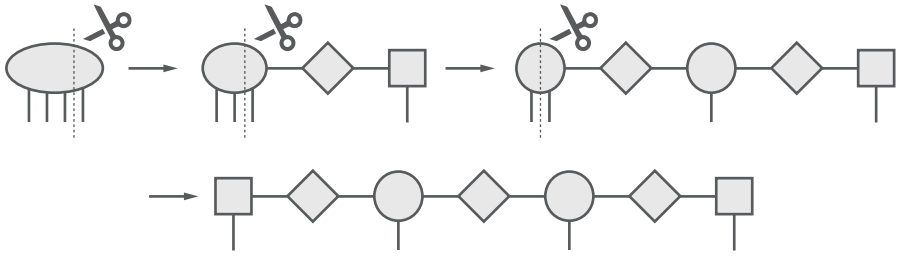
the number of independent coefficients $\psi_{ij\cdots k}$ in the basis expansion in general would be 2^n which quickly grows into a computationally unmanageable number as n increases. The goal is to find an alternative representation of $|\psi\rangle$ which is less data-intensive. We wish to write $|\psi\rangle$ as

$$|\psi\rangle = \sum_{ij\cdots k} \mathrm{Tr}(A_i^{[1]} A_j^{[2]} \cdots A_k^{[n]}) |ij\cdots k\rangle, \tag{66}$$

where $A_i^{[1]}, A_j^{[2]}, \ldots, A_k^{[n]}$ are indexed sets of matrices. Calculating the components of $|\psi\rangle$ then becomes a matter of calculating the products of matrices, hence the name *matrix product state*.

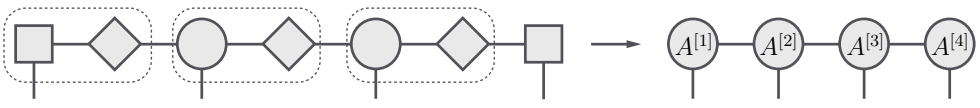
If the matrices are bounded in size, the representation becomes efficient in the sense that the amount of information required to describe them is only linear in n. The point of the method is to choose these matrices such that they provide a good (and compact) approximation to $|\psi\rangle$. For instance, if the matrices are at most χ by χ, the size of the representation scales as $nd\chi^2$, where d is the dimension of each subsystem.

Without loss of generality, we will now show how to obtain an MPS representation of an arbitrary four-party state $|\psi\rangle$. The key ingredient is the recursive application of the singular value decomposition (SVD) presented in Section 4. We start by considering the tensor which represents the state vector. We select a bipartition that separates one leg from the rest (starting at either end), and apply the SVD. This process is then repeated, traversing the entire tensor. This results in a 1D tensor network representation of the state, as shown below.



(67)

Finally the tensors are grouped—though the grouping has some ambiguity—resulting in the typical form shown below.



(68)

We note that one can recover the form in Eq. (66) as

$$|\psi\rangle = \sum_{ijkm} A_i^{[1]} A_j^{[2]} A_k^{[3]} A_m^{[4]} |ijkm\rangle. \tag{69}$$

In this case the first and last sets of matrices $A_i^{[1]}$ and $A_m^{[4]}$ have just a single row and a single column, respectively, so the product always yields a scalar and the trace is not required. The tensor network in Eq. (68) is called an MPS with open boundary conditions. MPSs also come with periodic boundary conditions, in which case the tensors form a ring instead of a chain, and the trace represents the contraction that closes the ring.

You might have noticed that we made a choice to perform the factorization starting from the left of the tensor and applying the SVD successively on tensors as we moved to the right. This apparent ambiguity has been characterized in detail [32]. For open boundary conditions as have been considered here, there is a canonical choice unique up to degeneracies in the spectrum of local reduced density operators [32].

For a general n-qubit quantum state it can be shown that the MPS matrix size can be bounded by $\chi = 2^{\lfloor n/2 \rfloor}$, which still grows exponentially as expected. A compact approximate representation is obtained by choosing a cutoff value ξ for the singular values across each partition, or a maximum number χ of singular values to be kept—see Example 13. This allows one to compress data by truncating the Hilbert space and is at the heart of the MPS algorithms. If an MPS is cut into two parts, the Schmidt rank of the decomposition, describing the degree of entanglement between the parts, is always $\leq \chi$, the dimension of the internal wire that was cut—see Example 10 for Schmidt rank and set $q = 0$ in Eq. (86) for the connection to entropy.

Above we have illustrated how to obtain the MPS representation of any pure quantum state, but it is normally not practical to factor states in this way for computational reasons. Instead, efficient MPS-generating algorithms are given an indirect, compact description of a state e.g. in the form of a nearest-neighbor Hamiltonian whose ground state we are interested in, and they then iteratively produce an MPS that closely approximates that state. The seminal algorithm of this type is the Density Matrix Renormalization Group (DMRG), which essentially works as a variational method in MPS space. Another class of algorithms can efficiently time-evolve MPSs under a nearest-neighbor Hamiltonian. One of the most used methods of this type is Time-Evolving Block Decimation (TEBD).

Example 13 (MPS approximation error)**.** As explained, matrix product state algorithms employ repeated application of the singular value decomposition. The size of the representation can be reduced by lossy truncation in one of two ways. Each of these rely on truncation of singular values. For a fixed rank, the Eckart-Young-Mirsky theorem—from the last Section—tells us that truncation of the singular values is the best approximation that one can expect.

In the first approximation, one can simply discard some fixed number of lowest singular values and their corresponding vectors—in other words, we will fix the dimension χ of all internal wires. In another approximation—which we will consider here—one will pick a cutoff value ξ, and truncate all singular values which are less than this. Such a cutoff value is not guaranteed to provide a useful partition of the singular values—e.g. it could be smaller than the smallest singular value. Here we will analyze the errors of this truncation assuming this cutoff partitions the singular values—which in practice is very often the case.

Given a bipartite state $|\psi\rangle$ we write

$$|\psi\rangle = \sum_{i=1}^{k} \sigma_i |u_i\rangle|v_i\rangle = \sum_{i=1}^{k} \sigma_i |i\rangle. \tag{70}$$

where to simplify the notation we write $|u_i\rangle|v_i\rangle$ as just $|i\rangle$. We order the singular values in

a non-decreasing sequence and introduce a cutoff $\xi > 0$:

$$0 < \sigma_1 \leq \sigma_2 \leq \sigma_3 \leq \cdots \leq \sigma_n \leq \xi \leq \sigma_{n+1} \leq \cdots \leq \sigma_k. \tag{71}$$

As a heuristic, the small cutoff ξ can be chosen such that $\xi < \frac{1}{\sqrt{\dim H}}$ where H is the vector space acted on by $|\psi\rangle$—you could also write $\xi < \frac{1}{\sqrt{d^q}}$ where d is the dimension of q constituent spaces.

And then we will partition our space in terms of the n singular values less than ξ and the $k - n$ ones that are greater than ξ.

$$|\psi\rangle = \sum_{i=1}^{n} \sigma_i |i\rangle + \sum_{j=n+1}^{k} \sigma_j |j\rangle = \sum_{i=1}^{n} \sigma_i |i\rangle + |\psi'\rangle, \tag{72}$$

where $|\psi'\rangle$ in (72) represents a (so far, non-normalized) approximation to $|\psi\rangle$. To understand the limits of validity of this approximation, we consider the inequality

$$\sum_{i=1}^{n} \sigma_i \leq n \cdot \sigma_n \leq n \cdot \xi, \tag{73}$$

and hence $n \cdot \sigma_n^2 \leq n \cdot \xi^2$.

Provided ξ is small, and for constant n we can return to (72) and consider another normalized but otherwise arbitrary vector $|\phi\rangle$ (element of the same space as $|\psi\rangle$ and $|\psi'\rangle$)

$$|\langle\phi|\psi - \psi'\rangle| = \left| \sum_{i=1}^{n} \sigma_i \langle\phi|i\rangle \right| \leq \sum_{i=1}^{n} \sigma_i |\langle\phi|i\rangle| \leq \sum_{i=1}^{n} \sigma_i \leq n \cdot \xi, \tag{74}$$

which scales linearly in the upper-bound of the error independent of $|\phi\rangle$.

It's common to introduce a function $\mathcal{O}(\xi)$ to collect terms up to a constant that approach zero no slower than $\mathcal{O}$'s argument. So (74) simply says that the absolute value of the difference between $|\psi'\rangle$ and $|\psi\rangle$ projected onto $\langle\phi|$ is bounded by a function proportional to ξ. From this we can readily conclude that the error in the approximation considered here is linear in ξ—or in other words, the error scales at most as $\mathcal{O}(\xi)$.

However, as $|\psi\rangle$ is normalized, we note that

$$\langle\psi|\psi\rangle = \sum_{i=1}^{n} \sigma_i^2 + \sum_{j=n+1}^{k} \sigma_j^2 = 1, \tag{75}$$

and then calculate the inner product

$$\langle\psi|\psi'\rangle = \sum_{j=n+1}^{k} \sigma_j^2 = 1 - \sum_{i=1}^{n} \sigma_i^2 \geq 1 - n\xi^2 = 1 + \mathcal{O}(\xi^2). \tag{76}$$

We hence conclude that the error in the inner product is only quadratic. In fact, if we normalize $|\psi'\rangle \to |\psi''\rangle$ and calculate

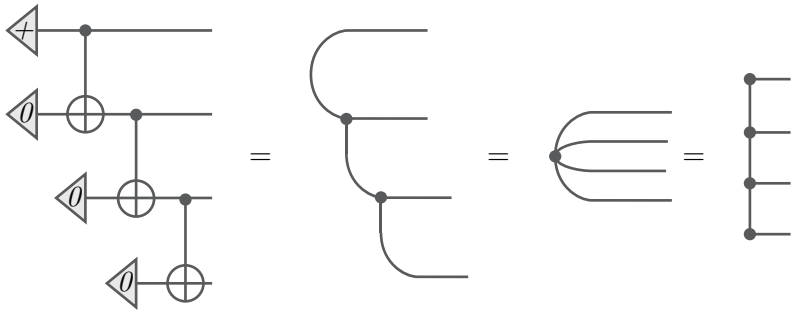
$$|\langle\psi|\psi''\rangle|^2 = 1 - \sum_{i=1}^{n} \sigma_i^2 \geq 1 - n\xi^2 = 1 + \mathcal{O}(\xi^2), \tag{77}$$

we recover the same thing.

Example 14 (MPS for the GHZ state)**.** The standard MPS representation of the Greenberger-Horne-Zeilinger (GHZ) state is given as

$$|\mathrm{GHZ}\rangle = \frac{1}{\sqrt{2}} \mathrm{Tr} \begin{pmatrix} |0\rangle & 0 \\ 0 & |1\rangle \end{pmatrix}^n = \frac{1}{\sqrt{2}}(|00\ldots0\rangle + |11\ldots1\rangle). \tag{78}$$

Alternatively, we may use a quantum circuit made of CNOT gates to construct the GHZ state, and then use the rewrite rules employed in Examples 4 and 8 to recover the familiar MPS comb-like structure consisting of COPY tensors:



(79)

Diagrammatically, any tensor network formed from connected COPY-tensors reduces to a single dot with the appropriate number of input and output legs. Hence one might write the n-party GHZ-state as

$$|\mathrm{GHZ}\rangle = \frac{1}{\sqrt{2}} \sum_{ijk\ldots l} \mathsf{COPY}^{ijk\ldots l} |ijk\ldots l\rangle. \tag{80}$$

Example 15 (MPS for the W state)**.** Like the GHZ state from Example 14, the n-qubit W state ($n \geq 3$) has the following MPS representation:

$$|W\rangle = \frac{1}{\sqrt{n}} \begin{pmatrix} |1\rangle & |0\rangle \end{pmatrix} \begin{pmatrix} |0\rangle & 0 \\ |1\rangle & |0\rangle \end{pmatrix}^{n-2} \begin{pmatrix} |0\rangle \\ |1\rangle \end{pmatrix} = \frac{1}{\sqrt{n}} \left(|10\ldots0\rangle + |010\ldots0\rangle + \ldots + |0\ldots01\rangle \right). \tag{81}$$

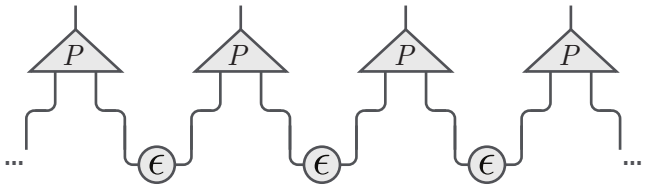
Example 16 (The AKLT model)**.** The AKLT model [33] (named after the authors Affleck, Kennedy, Lieb and Tasaki) is a theoretically important exactly solvable model of a spin-1 Heisenberg chain with an extra quadratic interaction term:

$$H = \sum_j \vec{S}_j \cdot \vec{S}_{j+1} + \frac{1}{3}(\vec{S}_j \cdot \vec{S}_{j+1})^2, \tag{82}$$

where $\vec{S}$ is a three-vector of the familiar spin-1 operators. The exact ground state of this Hamiltonian has an elegant expression as a matrix product state. Here we will carry on from Examples 1 and 7 which define and use the ϵ tensor—see also Example 8 which provides a quantum circuit realization for the corresponding singlet state. We start with a tensor product of these states,

(83)

and then project each neighboring qubit pair onto a three-dimensional (spin-1) Hilbert space using the projectors P:



(84)

The open wires on the P's have three degrees of freedom (spin-1). The projectors are defined as

$$P = |+\mathbf{1}\rangle\langle 11| + \frac{1}{\sqrt{2}}|\mathbf{0}\rangle(\langle 01| + \langle 10|) + |-\mathbf{1}\rangle\langle 00|. \tag{85}$$

The kets $|+\mathbf{1}\rangle$, $|\mathbf{0}\rangle$ and $|-\mathbf{1}\rangle$ are the standard spin-1 basis states and clearly the bond dimension $\chi = 2$. The AKLT state obtained after projection is rotationally symmetric but has a non-trivial entanglement structure and a host of other interesting properties—see e.g. [4] and the references therein.

The singular values found from the MPS factorization can be used to form a complete polynomial basis to express invariant quantities related to an MPS. This has a close connection to Schmidt rank and other concepts—see Examples 10 and 11.

Definition 17 (Rényi and von Neumann entropies)**.** Given a density operator ρ, its Rényi entropy of order q is defined to be

$$H_q(\rho) = \frac{1}{1-q}\log \operatorname{Tr}(\rho^q) = \frac{1}{1-q}\log \sum_i \lambda_i^q, \tag{86}$$

where λ_i are the eigenvalues of ρ. The case $q \to 0$ gives the rank of ρ (here we define $0^0 = 0$), and the limit $q \to 1$ recovers the familiar von Neumann entropy

$$H_{q\to 1}(\rho) = -\operatorname{Tr}(\rho \log \rho). \tag{87}$$

For a pure bipartite state $|\psi\rangle$ with subsystems A and B, the entropies of the reduced density operator $\rho_A = \operatorname{Tr}_B(|\psi\rangle\langle\psi|)$ can be used to quantify bipartite entanglement in the state. In this context they are called entanglement entropies. The eigenvalues of ρ_A (and ρ_B) are the squares of the Schmidt coefficients of $|\psi\rangle$, and $H_0(\rho_A)$ is equal to the Schmidt rank of the bipartition—see Example 10.

Area laws are quantified in terms of the scaling of entropy across tensor network partitions [34]. Whenever you bipartition a tensor network representing a pure quantum state, the total dimension χ of wires "cut" by this partition gives an upper bound to the entanglement entropy: $H_0(\rho_A) \leq \chi$ and $H_1(\rho_A) \leq \log \chi$.

6. COUNTING BY TENSOR CONTRACTION

In the tensor network language, counting problems can be expressed by evaluating fully contracted diagrams [1, 28, 35, 36].

To understand counting problems, imagine a phone book. But this phone book has a problem. The names are arranged randomly, without the standard alphabetical order we'd

all expect. If you're task is to determine Stephen Clark's phone number, the average number of names you'll need to examine before finding 'Stephen Clark' is exactly half the number in the phone book—assuming the name is unique. This is an example of a search problem. If we tell you the page number and location on the page where Stephen Clark's name appears, you can easily check and see if we're correct or not. Often times in computer science, problems are classified not in terms of how hard they are to solve—which is often unknown—but in terms of how hard it is to check if a given solution is correct or not.

Counting problems arise from search problems in the following way. Imagine you want to determine all the entries in this random phone book with the last name 'Jaksch'. This is then the counting version of the above search problem.

In these examples, both problems can be solved efficiently in the number of entries in the phone book. 'Efficiently' in the language of computer science means that the computational memory and runtime required is less than a polynomial in the problem size—in this case, the number of phone book entries.

In the language of computer science, more generally search problems are complete for the complexity class NP. Counting versions of NP-complete problems are #P-complete [37] in the language of computational complexity theory (pronounced sharp-P).

Connections between counting problems and tensor networks arose from the very early days of tensor networks. Penrose showed [1] that certain graph coloring problems can be solved by tensor contraction. We're going to outline Penrose's algorithm in Section 6 2. Before doing that, we will describe in Section 6 1 how tensor contractions can count the number of inputs that cause a given Boolean function to output 1.

1. Counting Boolean Formula Solutions

A Boolean function (a.k.a. switching function) takes an n-bit string of binary numbers—e.g. 00110101011—and maps this to a single binary digit (either 0 or 1). Several problems in physics can be mapped to Boolean functions [38]. In what follows, we will often denote this n-bit string as $\mathbf{x}$. We will show how to associate the Boolean function $f(\mathbf{x})$ with a non-normalized quantum state, written as a tensor network. We will establish the following (Remark 18) by a few examples.

Definition 18 (The class of Boolean tensors [35, 36])**.** Every Boolean function $f(\mathbf{x})$ gives rise to a Boolean tensor

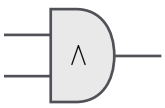
$$f = \sum_{\mathbf{x}} |f(\mathbf{x})\rangle\langle \mathbf{x}|, \tag{88}$$

with binary coefficients in $\{0, 1\}$. Moreover, a tensor network representing this state is determined from the classical logic gate network description of $f(\mathbf{x})$.

As an example, consider the logical AND operation which takes Boolean input variables x_1 and x_2 and outputs the Boolean product $x_1 \wedge x_2$—as a tensor, we write

$$\text{AND} := \sum_{x_1,x_2} |x_1 \wedge x_2\rangle\langle x_1, x_2| = |0\rangle\langle 00| + |0\rangle\langle 01| + |0\rangle\langle 10| + |1\rangle\langle 11|. \tag{89}$$

We will use the standard AND gate symbol also for the corresponding tensor:



(90)

Using (88) we can map any Boolean tensor f to an (unnormalized) Boolean state $|f\rangle$ by contracting the output with $|1\rangle$:

$$|f\rangle = f^\top|1\rangle = \sum_{\mathbf{x}} \langle f(\mathbf{x})|1\rangle|\mathbf{x}\rangle = \sum_{\mathbf{x}} f(\mathbf{x})|\mathbf{x}\rangle. \tag{91}$$

In addition to AND we have already seen other examples from this class of states, the GHZ and W states from Examples 14 and 15. The corresponding Boolean functions (for three variables/qubits) are

$$f_{\mathrm{GHZ}}(\mathbf{x}) = x_1x_2x_3 + (1-x_1)(1-x_2)(1-x_3) \tag{92}$$

and

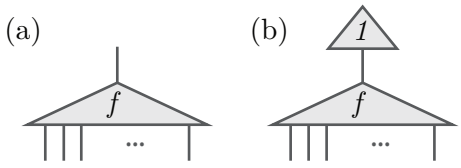
$$f_{\mathrm{W}}(\mathbf{x}) = x_1x_2(1-x_3) + x_1(1-x_2)x_3 + (1-x_1)x_2x_3. \tag{93}$$

Inserting these functions in Eq. (91), we obtain the corresponding (unnormalized) quantum states:

$$|\mathrm{GHZ}\rangle = |000\rangle + |111\rangle, \tag{94}$$

$$|\mathrm{W}\rangle = |001\rangle + |010\rangle + |100\rangle. \tag{95}$$

Now, given a Boolean function $f(\mathbf{x})$, each input for which the function returns 1 is said to satisfy the function. The number of satisfying inputs is then equal to the size of the support of $f(\mathbf{x})$.



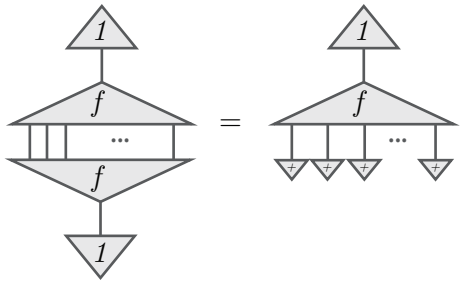
(96)

We may count the number of inputs that satisfy $f(\mathbf{x})$ by contracting the tensor f with $|1\rangle$ to obtain the corresponding Boolean state $|f\rangle$ (diagram (b)). The contraction post-selects the tensor network such that its support now consists solely of inputs that satisfy $f(\mathbf{x})$.

To explicitly translate this into a counting problem, we compute the squared norm,

$$\||f\rangle\|^2 = \langle f|f\rangle = \sum_{\mathbf{x},\mathbf{y}} f(\mathbf{x})f(\mathbf{y})\langle\mathbf{x}|\mathbf{y}\rangle = \sum_{\mathbf{x}} f(\mathbf{x})^2 = \sum_{\mathbf{x}} f(\mathbf{x}), \tag{97}$$

which clearly gives the number of satisfying inputs. In general for Boolean states the square of the two-norm always equals the one-norm since $f(\mathbf{x}) \in \{0,1\}$. In diagram form this is depicted as



(98)

where $|+\rangle := |0\rangle + |1\rangle$ and thus $|+\rangle \otimes |+\rangle \otimes \ldots \otimes |+\rangle = \sum_{\mathbf{x}} |\mathbf{x}\rangle$. More formally,

Theorem 19 (Counting SAT solutions [35, 36]). *Let f be a SAT instance. Then the standard two-norm length squared of the corresponding Boolean state $|f\rangle$ gives the number of satisfying assignments of the problem instance.*

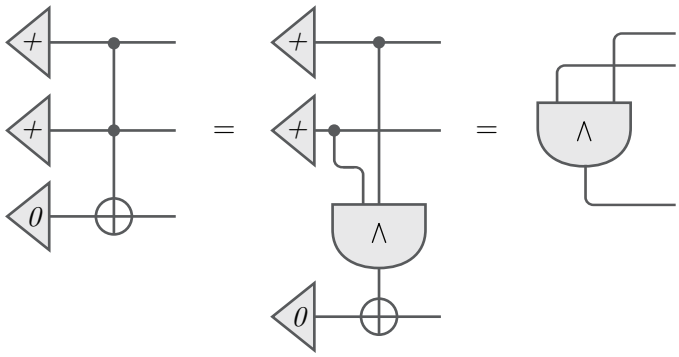
Solving the counting problem for general formula is known to be #P-complete [37]. Just to remind you what we talked about in the introduction to this section, computational complexity jargon for the set of the counting problems associated with the decision problems—decision problems seek to determine if a state is satisfiable at all, whereas counting problems seek to determine the total number of satisfying solutions. Indeed, the condition $\langle f|f\rangle > 0$ implies that the SAT instance f has a satisfying assignment. Determining whether this condition holds for general Boolean states is a NP-complete decision problem—as described in the introduction to this section.

Formulating these problems as tensor contractions allows the adaptation of tools developed to simulate quantum systems and circuits such that they now apply to an areas traditionally considered in computer science [35, 36]. We adapted these tools and discovered efficient tensor network descriptions of finite Abelian lattice gauge theories [27]. These tools also lead to the discovery of a wide class of efficiently contractable tensor networks, representing counting problems [36].

Example 20 (AND from Toffoli). The Toffoli gate has long been studied in reversible computing, and also in quantum computing. One can view this operation as being formed internally by an AND tensor, two copy tensors (black dots) and one XOR tensor ($\oplus$)—see Examples 3 and 14. One can create the $|\text{AND}\rangle$ state in an experiment by preparing the state

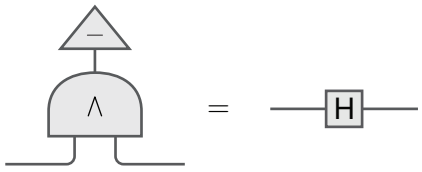
$$|+\rangle|+\rangle|0\rangle, \tag{99}$$

where $|+\rangle := |0\rangle + |1\rangle$, and then applying the Toffoli gate. This is illustrated with the following tensor diagram.



(100)

Example 21 (Hadamard from AND). By considering the contraction formed with the state $|-\rangle := \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ and the output of the AND tensor, one recovers the Hadamard gate defined in Eq. (21). Note that we've been using $|+\rangle$ without normalization and here we normalize $|-\rangle$.

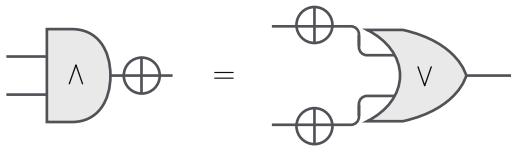


(101)

Example 22 (De Morgan's laws)**.** A common identity used in Boolean algebra is

$$\neg(a \wedge b) = (\neg a) \vee (\neg b), \tag{102}$$

where negation denoted as $\neg$, logical OR as $\vee$, and logical AND as $\wedge$. When expressed as a tensor network,



(103)

this relation has the same structure as the relationship between the COPY and XOR tensors and the Hadamard gate in Eq. (26) Such diagrammatic rewrite rules can be used to formalize a system of graphical reasoning, and reduce calculations through easily employed graphical rewrite identities.

The synthesis problem seeks to determine how to build a logical circuit from basic logic gates (such as AND) that realizes a given Boolean function. Given this logical circuit, we can obtain the corresponding tensor network simply by replacing the gates with their tensor counterparts. Using tensors that represent classical logical gates provides an alternative means to determine tensor networks representing for instance the GHZ and AND states [25], compared to the MPS representations given in Examples 14 and 15.

2. Counting Graph Colorings

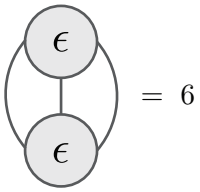
Given a 3-regular planar graph[8], how many possible edge colorings using three colors exist, such that all edges connected to each node have distinct colors? This counting problem can be solved in an interesting (if not computationally efficient) way using the order-3 ϵ tensor, which is defined in terms of components as

$$\begin{aligned} \epsilon_{012} = \epsilon_{120} = \epsilon_{201} = 1, \\ \epsilon_{021} = \epsilon_{210} = \epsilon_{102} = -1, \end{aligned} \tag{104}$$

otherwise zero. The counting algorithm is stated as

Theorem 23 (Planar graph 3-colorings, Penrose 1971 [1])**.** *The number K of proper 3-edge-colorings of a planar 3-regular graph is obtained by replacing each node with an order-3 epsilon tensor, replacing each edge with a wire, and then contracting the resulting tensor network.*

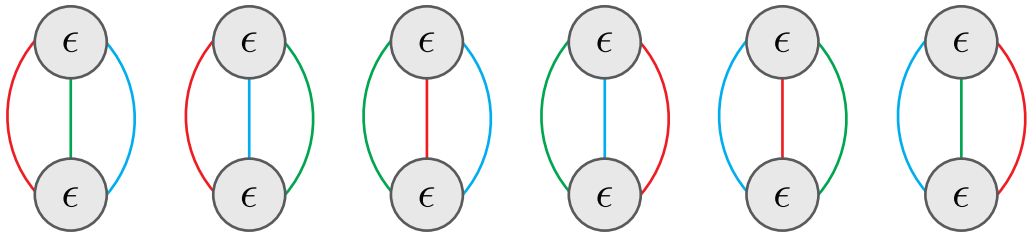
We will first consider the simplest case, a graph with just two nodes. In this case we obtain



(105)

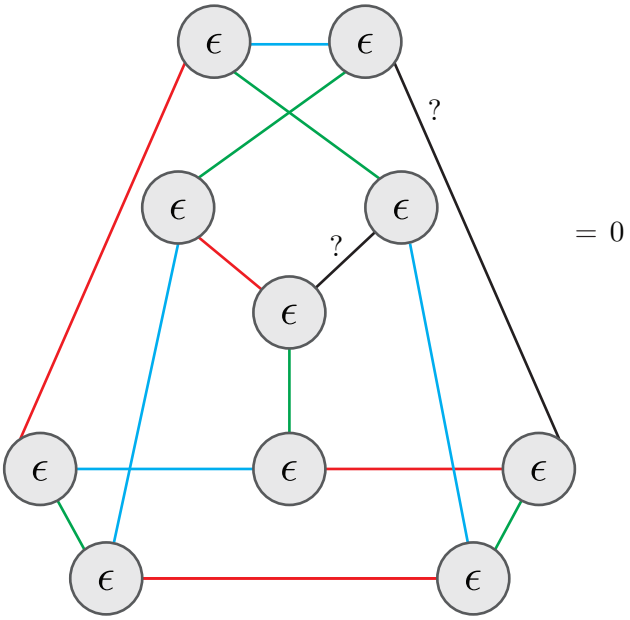
[8] A graph is k-regular iff every node has exactly k edges connected to it.

There are indeed 6 distinct edge colorings for this graph, given as



(106)

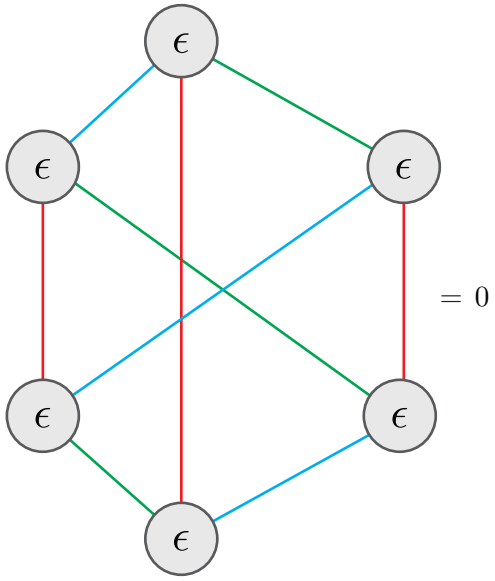
To understand Theorem 23, note first that the contraction K of the epsilon tensor network is the sum of all possible individual assignments of the index values to the epsilon tensors comprising the network. Each of the three possible index values can be understood as a color choice for the corresponding edge. Whenever the index values for a given epsilon tensor are not all different, the corresponding term in K is zero. Hence only allowed color assignments result in nonzero contributions to K, and for a graph that does not admit a proper 3-edge-coloring we will have $K = 0$. For instance, for the non-3-colorable Petersen graph we obtain



(107)

However, for K to actually equal the number of allowed colorings, each nonzero term must have the value 1 (and not -1). This is only guaranteed if the graph is planar, as can

be seen by considering the non-planar graph $K_{3,3}$:



(108)

The edges can be colored with three colors—in 12 different ways—yet the contraction vanishes.

The computational complexity of this problem has been studied in [40]. Interesting, by a well known result (Heawood 1897), the 3-colorings as stated above, are one quarter of the ways of coloring the faces of the graph with four colors, so that no two like-colored faces have an edge in common.

Example 24 (Physical implementation of ϵ_{abc} in quantum computing)**.** In quantum computing, typically one works with qubits (two level quantum systems) but implementations using qutrits exist (three level quantum systems, available in e.g. nitrogen vacancy centers in diamond—see for instance [41]). The epsilon tensor ϵ_{abc} could be realized directly as a locally invariant 3-party state using qutrits, and can also be embedded into a qubit system. We leave it to the reader to show that by pairing qubits, ϵ_{abc} can be represented with six qubits, where each leg now represents a qubit pair. (Note that a basis of 3 states can be isometrically embedded in 4-dimensional space in any number of ways.) Show further that the construction can be done such that the two qubit pairs (together representing one leg) are symmetric under exchange. Note the similarities with Eq. (85) in Example 16.

7. FRONTIERS IN TENSOR NETWORKS

Tensor network methods represent a vibrant area of active research, with new results and ideas appearing regularly. Here we have covered the elementary aspects of the tensor network language and three applications. The first was the matrix product state representation, the next was tensor contractions to count Boolean formula solutions and the final application focused on tensor contractions to evaluate 3-edge-colorings of 3-regular planar graphs. These three sample applications should provide a good base to move forward into active research.

The most common tensor network structures and algorithms used in quantum mechanics include Matrix Product States (MPS) [42–44] (see Section 5) and the related Density Matrix Renormalization Group (DMRG) [8], Matrix Product Operators (MPO) [45], Tensor

Trains [15], Tree Tensor Networks (TTN) [46], the Multiscale Entanglement Renormalization Ansatz (MERA) [5, 47–49], Projected Entangled Pair States (PEPS) [50], Correlator Product States (CPS) [51] and Time-Evolving Block Decimation (TEBD) [52]—see also time-evolution with MERA [53, 54].

Our reference list above is admittedly very much incomplete but should provide a solid starting place for further study. Going further, there are several reviews that we encourage readers to consult. These proceed largely towards approaches that map lattice problems to tensor networks as tools to solve models of strongly correlated systems [4–8, 10, 11, 14]—see also the viewpoint [9]. There is a growing and active community exploring the use of tensor network algorithms as a means to discover and understand new properties of quantum systems.

In terms of the graphical language, category theory is a branch of mathematics well suited to describe a wide range of networks [18]. Quantum circuits were first given a 'categorical model' in pioneering work by Lafont in 2003 [55] and dagger compact closed categories [56], also called Baez-Dolan †-categories, were first derived to describe both standard quantum theory as well as classes of topological quantum field theories in seminal work published in 1995 [56]. (See [18] for a well written review of categorical quantum mechanics.) For practical purposes, the graphical language turns out to be mathematically equivalent to the categorical formulation.

There has been some recent excitement surrounding MERA [5, 47–49]—which is capable of representing the ground state of certain many-body models at their critical points—and its connection to quantum gravity research. Several interesting discoveries [57] have recently been made around the so called tensor network incarnation of the AdS/MERA correspondence; networks which realize a discrete anti-de Sitter space have a corresponding MERA network which represents the ground state of a critical system [57, 58]. This has generated significant recent interest and excitement. If this sounds exciting to you, you might want to take a look at [11, 13, 59].

ACKNOWLEDGMENTS

We've had a number of excellent collaborators over the years, who certainly influenced our understanding of the topic. The least we can do is thank them here. In alphabetical order we thank John Baez, Stephen Clark, Sam Denny, Dieter Jaksch, Tomi Johnson, Marco Lanzagorta, Jason Morton, Lea Trenkwalder, Jacob Turner, Chris Wood and Zoltán Zimborás, as well as others we're probably forgetting. J.B. acknowledges AFOSR grant FA9550-16-1-0300, Models and Protocols for Quantum Distributed Computation, for financial support. Diagrams and cover courtesy of Lusa Zheglova (illustrator).

[1] Roger Penrose, "Applications of negative dimensional tensors," in *Combinatorial Mathematics and its Applications*, edited by D. Welsh (Academic Press, New York, 1971) pp. 221–244.

[2] D. Deutsch, "Quantum computational networks," Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences **425**, 73–90 (1989).

[3] Richard P. Feynman, "Quantum mechanical computers," Foundations of Physics **16**, 507–531 (1986).

[4] R. Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states," Annals of Physics **349**, 117–158 (2014), arXiv:1306.2164.

[5] G. Vidal, "Entanglement renormalization: an introduction," in *Understanding Quantum Phase Transitions*, edited by Lincoln D. Carr (Taylor & Francis, Boca Raton, 2010).

[6] F. Verstraete, V. Murg, and J. I. Cirac, "Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems," Advances in Physics **57**, 143–224 (2008), arXiv:0907.2796.

[7] J. I. Cirac and F. Verstraete, "Renormalization and tensor product states in spin chains and lattices," J. Phys. A Math. Theor. **42**, 504004 (2009), arXiv:0910.1130.

[8] U. Schollwöck, "The density-matrix renormalization group in the age of matrix product states," Annals of Physics **326**, 96–192 (2011), arXiv:1008.3477.

[9] S. Sachdev, "Viewpoint: Tensor networks—a new tool for old problems," Physics **2**, 90 (2009), arXiv:1006.0675.

[10] Ulrich Schollwöck, "The density-matrix renormalization group: a short introduction," Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences **369**, 2643–2661 (2011).

[11] R. Orús, "Advances on tensor network theory: symmetries, fermions, entanglement, and holography," European Physical Journal B **87**, 280 (2014), arXiv:1407.6552.

[12] J. Eisert, "Entanglement and tensor network states," Modeling and Simulation **3**, 520 (2013), arXiv:1308.3318.

[13] G. Evenbly and G. Vidal, "Tensor Network States and Geometry," Journal of Statistical Physics **145**, 891–918 (2011), arXiv:1106.1082.

[14] J. C. Bridgeman and C. T. Chubb, "Hand-waving and Interpretive Dance: An Introductory Course on Tensor Networks," ArXiv e-prints (2016), arXiv:1603.03039 [quant-ph].

[15] 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**, 249–429 (2016), 1609.00893.

[16] Andrzej Cichocki, Anh-Huy Phan, Qibin Zhao, Namgil Lee, Ivan Oseledets, Masashi Sugiyama, and Danilo P. Mandic, "Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives," Foundations and Trends in Machine Learning **9**, 431–673 (2017).

[17] M. J. Hartmann, J. Prior, S. R. Clark, and M. B. Plenio, "Density matrix renormalization group in the Heisenberg picture," Phys. Rev. Lett. **102**, 057202 (2009), arXiv:0808.0666.

[18] J. C. Baez and A. Lauda, "A Prehistory of n-Categorical Physics," in *Deep Beauty* (Cambridge University Press, 2011) pp. 13–128, arXiv:0908.2469.

[19] Jacob D. Biamonte, Ville Bergholm, and Marco Lanzagorta, "Tensor network methods for invariant theory," J. Phys. A: Math. Theor. **46**, 475301 (2013), arXiv:1209.0631.

[20] A. Critch and J. Morton, "Algebraic Geometry of Matrix Product States," SIGMA **10**, 095 (2014), arXiv:1210.2812 [quant-ph].

[21] William K. Wootters, "Entanglement of formation of an arbitrary state of two qubits," Phys. Rev. Lett. **80**, 2245–2248 (1998), arXiv:quant-ph/9709029.

[22] V. Coffman, J. Kundu, and W. K. Wootters, "Distributed entanglement," Phys. Rev. A **61**, 052306 (2000), arXiv:quant-ph/9907047.

[23] Julia Kempe, "Multiparticle entanglement and its applications to cryptography," Phys. Rev. A **60**, 910–916 (1999), arXiv:quant-ph/9902036.

[24] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter, "Elementary gates for quantum computation," Phys. Rev. A **52**, 3457–3467 (1995), arXiv:quant-ph/9503016.

[25] J. D. Biamonte, S. R. Clark, and D. Jaksch, "Categorical tensor network states," AIP Advances **1**, 042172 (2011).

[26] Ville Bergholm and Jacob D. Biamonte, "Categorical quantum circuits," J. Phys. A: Math. Theor. **44**, 245304 (2011), arXiv:1010.4840.

[27] S. J. Denny, J. D. Biamonte, D. Jaksch, and S. R. Clark, "Algebraically contractible topological tensor network states," Journal of Physics A Mathematical General **45**, 015309 (2012), arXiv:1108.0888.

[28] Roger Penrose, "The theory of quantized directions," in *Collected Works*, Vol. 1 (October 1953–67) (Oxford University Press, 2010) Chap. 31, pp. 769–800.

[29] Christopher J. Wood, Jacob D. Biamonte, and David G. Cory, "Tensor networks and graphical calculus for open quantum systems," Quant. Inf. Comp. **15**, 759–811 (2015), arXiv:1111.6950.

[30] S. Meznaric and J. Biamonte, "Tensor networks for entanglement evolution," in *Quantum Information and Computation for Chemistry: Advances in Chemical Physics*, Vol. 154, edited by Sabre Kais (John Wiley & Sons, 2014) pp. 561–574, arXiv:1204.3599.

[31] M. B. Hastings, "An area law for one-dimensional quantum systems," Journal of Statistical Mechanics: Theory and Experiment **8**, 08024 (2007), arXiv:0705.2024.

[32] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac, "Matrix product state representations," Quant. Inf. and Comp. **7**, 401–430 (2007), arXiv:quant-ph/0608197.

[33] I. Affleck, T. Kennedy, E. H. Lieb, and H. Tasaki, "Rigorous results on valence-bond ground states in antiferromagnets," Phys. Rev. Lett. **59**, 799–802 (1987).

[34] J. Eisert, M. Cramer, and M. B. Plenio, "Colloquium: Area laws for the entanglement entropy," Reviews of Modern Physics **82**, 277–306 (2010), arXiv:0808.3773.

[35] T. H. Johnson, J. D. Biamonte, S. R. Clark, and D. Jaksch, "Solving search problems by strongly simulating quantum circuits," Scientific Reports **3**, 1235 (2013), arXiv:1209.6010.

[36] Jacob Biamonte, Jason Morton, and Jacob Turner, "Tensor network contractions for #SAT," Journal of Statistical Physics **160**, 1389–1404 (2015), arXiv:1405.7375.

[37] Leslie G Valiant, "The complexity of computing the permanent," Theoretical computer science **8**, 189–201 (1979).

[38] J. D. Whitfield, M. Faccin, and J. D. Biamonte, "Ground-state spin logic," EPL (Europhysics Letters) **99**, 57004 (2012), arXiv:1205.1742 [quant-ph].

[39] Jacob Biamonte, "Charged string tensor networks," Proceedings of the National Academy of Sciences **114**, 2447 (2017).

[40] Mingji Xia, Peng Zhang, and Wenbo Zhao, "Computational complexity of counting problems on 3-regular planar graphs," Theoretical Computer Science **384**, 111–125 (2007).

[41] F. Dolde *et al.*, "High-fidelity spin entanglement using optimal control," Nature Communications **5**, 3371 (2014), arXiv:1309.4430.

[42] M. Fannes, B. Nachtergaele, and R. F. Werner, "Finitely correlated states on quantum spin chains," Communications in Mathematical Physics **144**, 443–490 (1992).

[43] Stellan Östlund and Stefan Rommer, "Thermodynamic limit of density matrix renormalization," Phys. Rev. Lett. **75**, 3537–3540 (1995), arXiv:cond-mat/9503107.

[44] Stefan Rommer and Stellan Östlund, "Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group," Phys. Rev. B **55**, 2164–2181 (1997), arXiv:cond-mat/9606213.

[45] F. Verstraete, J. J. García-Ripoll, and J. I. Cirac, "Matrix product density operators: Simulation of finite-temperature and dissipative systems," Phys. Rev. Lett. **93**, 207204 (2004), arXiv:cond-mat/0406426.

[46] Y.-Y. Shi, L.-M. Duan, and G. Vidal, "Classical simulation of quantum many-body systems with a tree tensor network," Phys. Rev. A **74**, 022320 (2006), arXiv:quant-ph/0511070.

[47] G. Vidal, "Entanglement renormalization," Phys. Rev. Lett. **99**, 220405 (2007), arXiv:cond-mat/0512165.

[48] V. Giovannetti, S. Montangero, and R. Fazio, "Quantum Multiscale Entanglement Renormalization Ansatz Channels," Physical Review Letters **101**, 180503 (2008), arXiv:0804.0520.

[49] G. Vidal, "Class of quantum many-body states that can be efficiently simulated," Phys. Rev. Lett. **101**, 110501 (2008), arXiv:quant-ph/0610099.

[50] F. Verstraete, M. M. Wolf, D. Perez-Garcia, and J. I. Cirac, "Criticality, the area law, and the computational power of projected entangled pair states," Physical Review Letters **96**, 220601 (2006), arXiv:quant-ph/0601075.

[51] S. Al-Assam, S. R. Clark, C. J. Foot, and D. Jaksch, "Capturing long range correlations in two-dimensional quantum lattice systems using correlator product states," Phys. Rev. B **84**, 205108 (2011), arXiv:1107.0936.

[52] G. Vidal, "Efficient classical simulation of slightly entangled quantum computations," Physical Review Letters **91**, 147902 (2003), arXiv:quant-ph/0301063.

[53] M. Rizzi, S. Montangero, and G. Vidal, "Simulation of time evolution with multiscale entanglement renormalization ansatz," Phys. Rev. A **77**, 052328 (2008), arXiv:0706.0868.

[54] J. Molina-Vilaplana and J. Prior, "Entanglement, tensor networks and black hole horizons," General Relativity and Gravitation **46**, 1823 (2014), arXiv:1403.5395.

[55] Yves Lafont, "Towards an algebraic theory of boolean circuits," Journal of Pure and Applied Algebra **184**, 2003 (2003).

[56] John C. Baez and James Dolan, "Higher-dimensional algebra and topological quantum field theory," Journal of Mathematical Physics **36**, 6073–6105 (1995).

[57] B. Swingle, "Entanglement renormalization and holography," Phys. Rev. D **86**, 065007 (2012), arXiv:0905.1317.

[58] N. Bao, C. Cao, S. M. Carroll, A. Chatwin-Davies, N. Hunter-Jones, J. Pollack, and G. N. Remmen, "Consistency conditions for an AdS multiscale entanglement renormalization ansatz correspondence," Phys. Rev. D **91**, 125036 (2015), arXiv:1504.06632.

[59] M. Van Raamsdonk, "Lectures on gravity and entanglement," (2016), arXiv:1609.00026 [hep-th].

Appendix A: Tensors and Tensor Products

The definition of a tensor starts with the *tensor product* $\otimes$. There are many equivalent ways to define it, but perhaps the simplest one is through basis vectors. Let V and W be finite-dimensional vector spaces over the same field of scalars $\mathbb{K}$. In physics-related applications $\mathbb{K}$ is typically either the real numbers $\mathbb{R}$ or the complex numbers $\mathbb{C}$. Now $V \otimes W$ is also a vector space over $\mathbb{K}$. If V and W have the bases $\{e_j\}_j$ and $\{f_k\}_k$, respectively, the symbols $\{e_j \otimes f_k\}_{jk}$ form a basis for $V \otimes W$. Thus, for finite-dimensional spaces $\dim(V \otimes W) = \dim V \, \dim W$.

The tensor product of two individual vectors $v \in V$ and $w \in W$ is denoted as $v \otimes w$. For vectors the tensor product is a bilinear map $V \times W \to V \otimes W$, i.e. one that is linear in both input variables. For finite-dimensional spaces one can obtain the standard basis coordinates of the tensor product of two vectors as the *Kronecker product* of the standard basis coordinates of the individual vectors:

$$(v \otimes w)^{jk} = v^j w^k. \tag{A1}$$

It is important to notice that due to the bilinearity $\otimes$ maps many different pairs of vectors (v, w) to the same product vector: $v \otimes (sw) = (sv) \otimes w = s(v \otimes w)$, where $s \in \mathbb{K}$. For inner product spaces (such as the Hilbert spaces encountered in quantum mechanics) the tensor product space inherits the inner product from its constituent spaces:

$$\langle v_1 \otimes w_1, v_2 \otimes w_2 \rangle_{V \otimes W} = \langle v_1, v_2 \rangle_V \langle w_1, w_2 \rangle_W . \tag{A2}$$

A *tensor* T is an element of the tensor product of a finite number of vector spaces over a common field of scalars $\mathbb{K}$. The dual space V^* of a vector space V is defined as the space of linear maps from V to $\mathbb{K}$. It is not hard to show that V^* is a vector space over $\mathbb{K}$ on its own. This leads us to define the concept of an order-(p, q) tensor, an element of the tensor product of p primal spaces and q dual spaces:

$$T \in W_1 \otimes W_2 \otimes \ldots \otimes W_p \otimes V_1^* \otimes V_2^* \otimes \ldots \otimes V_q^*. \tag{A3}$$

Given a basis $\{e^{(i)}{}_k\}_k$ for each vector space W_i and a dual basis $\{\eta^{(i)k}\}_k$ for each dual space V_i^*, we may expand T in the tensor products of these basis vectors:

$$T = T^{i_1 \ldots i_p}{}_{j_1 \ldots j_q} \, e^{(1)}{}_{i_1} \otimes \ldots \otimes e^{(p)}{}_{i_p} \otimes \eta^{(1) j_1} \otimes \ldots \otimes \eta^{(q) j_q}. \tag{A4}$$

$T^{i_1 \ldots i_p}{}_{j_1 \ldots j_q}$ is simply an array of scalars containing the basis expansion coefficients. Here we have introduced the *Einstein summation convention*, in which any index that is repeated exactly twice in a term, once up, once down, is summed over. This allows us to save a considerable number of sum signs, without compromising on the readability of the formulas. Traditionally basis vectors carry a lower (covariant) index and dual basis vectors an upper (contravariant) index.

A tensor is said to be *simple* if it can be written as the tensor product of some elements of the underlying vector spaces: $T = v^{(1)} \otimes \ldots \otimes v^{(q)} \otimes \varphi^{(1)} \otimes \ldots \otimes \varphi^{(p)}$. This is not true for most tensors; indeed, in addition to the bilinearity, this is one of the properties that separates tensors from mere Cartesian products of vectors. However, any tensor can be written as a linear combination of simple tensors, e.g. as in Eq. (A4).

For every vector space W there is a unique bilinear map $W \otimes W^* \to \mathbb{K}$, $w \otimes \phi \mapsto \phi(w)$ called a natural pairing, where the dual vector maps the primal vector to a scalar. One can apply this map to any pair of matching primal and dual spaces in a tensor. It is called a *contraction* of the corresponding upper and lower indices. For example, if we happen to have $W_1 = V_1$ we may contract the corresponding indices on T:

$$\begin{aligned} C_{1,1}(T) &= T^{i_1 \ldots i_p}{}_{j_1 \ldots j_q}\, \eta^{(1)j_1}(e^{(1)}{}_{i_1})\ e^{(2)}{}_{i_2} \otimes \ldots \otimes e^{(p)}{}_{i_p} \otimes \eta^{(2)j_2} \otimes \ldots \otimes \eta^{(q)j_q} \\ &= T^{k\, i_2 \ldots i_p}{}_{k\, i_2 \ldots j_q}\, e^{(2)}{}_{i_2} \otimes \ldots \otimes e^{(p)}{}_{i_p} \otimes \eta^{(2)j_2} \otimes \ldots \otimes \eta^{(q)j_q}, \end{aligned} \tag{A5}$$

since the defining property of a dual basis is $\eta^{(1)j_1}(e^{(1)}{}_{i_1}) = \delta^{j_1}{}_{i_1}$. Hence the contraction eliminates the affected indices (k is summed over), lowering the tensor order by $(1,1)$.

We can see that an order-$(1,0)$ tensor is simply a vector, an order-$(0,1)$ tensor is a dual vector, and can define an order-$(0,0)$ tensor to correspond to a plain scalar. But what about general, order-(p,q) tensors? How should they be understood? Using contraction, they can be immediately reinterpreted as multilinear maps from vectors to vectors:

$$\begin{aligned} T' :&\quad V_1 \otimes \ldots \otimes V_q \to W_1 \otimes \ldots \otimes W_p, \\ T'(v^{(1)} \otimes \ldots \otimes v^{(q)}) &= T^{i_1 \ldots i_p}{}_{j_1 \ldots j_q}\, e^{(1)}{}_{i_1} \otimes \ldots \otimes e^{(p)}{}_{i_p} \times \eta^{(1)j_1}(v^{(1)}) \times \ldots \times \eta^{(q)j_q}(v^{(q)}), \end{aligned} \tag{A6}$$

where we tensor-multiply T and the vectors to be mapped together, and then contract the corresponding indices. However, this is not the only possible interpretation. We could just as easily see them as mapping dual vectors to dual vectors:

$$\begin{aligned} T'' :&\quad W_1^* \otimes \ldots \otimes W_p^* \to V_1^* \otimes \ldots \otimes V_q^*, \\ T''(\varphi^{(1)} \otimes \ldots \otimes \varphi^{(p)}) &= T^{i_1 \ldots i_p}{}_{j_1 \ldots j_q}\, \varphi^{(1)}(e^{(1)}{}_{i_1}) \times \ldots \times \varphi^{(p)}(e^{(p)}{}_{i_p}) \times \eta^{(1)j_1} \otimes \ldots \otimes \eta^{(q)j_q}. \end{aligned} \tag{A7}$$

Essentially we may move any of the vector spaces to the other side of the arrow by taking their dual:

$$W \otimes V^* \quad \cong \quad \mathbb{K} \to W \otimes V^* \quad \cong \quad V \to W \quad \cong \quad V \otimes W^* \to \mathbb{K} \quad \cong \quad W^* \to V^*, \tag{A8}$$

where all the arrows denote linear maps. Any and all input vectors are mapped to scalars by the corresponding dual basis vectors in expansion (A4), whereas all input dual vectors map the corresponding primal basis vectors to scalars.

If we expand the input vectors $v^{(k)}$ in Eq. (A6) using the same bases as when expanding the tensor T, we obtain the following equation for the expansion coefficients:

$$T'(v^{(1)} \otimes \ldots \otimes v^{(q)})^{i_1 \ldots i_p} = T^{i_1 \ldots i_p}{}_{j_1 \ldots j_q}\, v^{(1)j_1} \cdots v^{(q)j_q}. \tag{A9}$$

This is much less cumbersome than Eq. (A6), and contains the same information. This leads us to adopt the *abstract index notation* for tensors, in which the indices no longer denote the components of the tensor in a particular basis, but instead signify the tensor's order. Tensor products are denoted by simply placing the tensor symbols next to each other. Within each term, any repeated index symbol must appear once up and once down, and denotes contraction over those indices. Hence, x^a denotes a vector (with one contravariant index), ω_a a dual vector (with one covariant index), and $T^{ab}{}_c$ an order-$(2,1)$ tensor with

two contravariant and one covariant indices. $S^{ab}{}_{cde}x^c y^d P^e{}_a$ denotes the contraction of an order-$(2,3)$ tensor S, an order-$(1,1)$ tensor P, and two vectors, x and y, resulting in an order-$(1,0)$ tensor with one uncontracted index, b.

In many applications, for example in differential geometry, the vector spaces associated with a tensor are often copies of the same vector space V or its dual V^*, which means that any pair of upper and lower indices can be contracted, and leads to the tensor components transforming in a very specific way under basis changes. This specific type of a tensor is called an order-(p,q) tensor *on the vector space* V. However, here we adopt a more general definition, allowing $\{V_k\}_k$ and $\{W_k\}_k$ to be all different vector spaces.