
Approximate tensor network contraction

The authors are grateful to Glen Evenbly, Johnnie Gray, Daniel Malz, and Ashley Milsted for reviewing this chapter.

Rough overview (in words)

Tensor network algorithms are a versatile tool that is playing an increasingly important role in problems both within and outside of physics and quantum computation [150], whenever the size of the underlying linear space is exponentially large in some appropriately defined dimension (i.e., tensor decomposition of the space). Their application to exponentially large linear systems is ultimately limited by the ability to contract (i.e., sum over repeated indices) large networks of tensors, in particular when the network forms a graph with many loops. Quantum approximate contraction of tensor networks [54] is a quantum algorithm for contracting arbitrary tensor networks up to a constant additive error. Estimating partition functions up to an additive error is a special case of the general problem, where all elements of the tensor network are positive.

This quantum approach to approximate tensor network contraction is of particular interest since many commercially relevant problems do not care about asymptotic speedups, but rather time-to-solution on smaller or medium problem sizes, and oftentimes approximate solutions found with heuristics are good enough. Tensor network (sometimes called quantum-inspired) algorithms for industrially relevant problems can be used heuristically, and the quantum approximate contraction backend might be used in cases where the classical algorithms do not provide sufficient accuracy, speed, or scale. Quantum-inspired classical algorithms based on tensor networks might allow for the identification of promising heuristic applications of quantum computing.

At this time, however, the only known problems where the quantum backend provides substantial speedup is for problems originating from quantum computing itself, such as quantum computational supremacy experiments based on random quantum circuits [826].

Rough overview (in math)

We define a tensor network as an abstract object $T(G, M)$ defined on a graph $G = (V, E)$, where to each vertex $v \in V$ we associate a tensor $M^{(v)}$ with one index for each adjacent edge. The tensor network $T(G, M)$ is closed, in that all edges are contracted. This means that for any specific set of tensors M on G , $T(G, M)$ maps to a scalar. Given the graph G , we define a contraction pathway as a sequence of edges in G , where at each step the pair of vertices adjacent to the edge are merged into one vertex, until, at the end of the sequence, only one vertex remains in the graph. When merging two vertices, it can occur that edges that were previously distinct now coincide. It is important to keep track of the multiplicity of such edges. The optimal contraction pathway, in terms of cost scaling, is the merge sequence for which the maximum number of edges (counting multiplicity) incident to any vertex at any of the steps—a quantity called the contraction width—is minimized. The optimal contraction width is related to the tree width of the graph [741]. Classical exact contraction algorithms typically scale exponentially in the contraction width [330, 911, 483, 455]. For generic, loopy networks, the contraction width is expected to be polynomially related to $|V|$; thus, the exact contraction algorithm will quickly become intractable with growing $|V|$. However, many approximate contraction methods exist [816, 454].

Suppose we fix a contraction pathway for which the sequence of edges forms a path in G , known as a “bubbling” because one can draw a bubble around contracted vertices, which sequentially expands as the path is traversed. This may not be the pathway with optimal contraction width (for which the sequence of edges need not always form a path). Then, for any $\epsilon > 0$, there exists a quantum algorithm that runs in $O(|V|\epsilon^{-2}\text{poly}(q^d))$ quantum time and outputs a complex number r such that [54]

$$\Pr(|T(G, M) - r| \geq \epsilon\Delta) \leq \frac{1}{4},$$

where d is the maximum degree of the graph and q is the dimension of the edge Hilbert space (or bond dimension). The parameter Δ is the sequential norm of the operations in the contraction path: $\Delta = \prod_{v \in V} \|O_v\|$, where O_v are called swallowing operators (see Definitions 3.1 and 3.2 in [54]), which control the sequential contraction of the tensor network.

Intuitively, one can think of contracting the network one edge at a time along a connected path, such as a snake covering a 2D lattice. At each step of the way, the contracted vertices—which form a potentially large tensor—are encoded as a quantum state, and each new vertex is contracted by a local operator O_v (the process is called bubbling in [54]). The dimension of the “state” can increase or decrease with every operation. Each operator O_v in the contraction pathway is approximately mapped onto a unitary operator on the linear space (q^d dimensional) connecting vertex v in the network plus one ancilla qubit. The approximation comes from the Solovay–Kitaev theorem. This way, the exact contraction of the tensor network is approximately mapped onto a quantum circuit of volume roughly equal to the graph “volume.” The output state of the quantum circuit encodes the result of the tensor network contraction into one of its amplitudes. In [54], they show how to estimate this amplitude using the Hadamard test, contributing the factor of ϵ^{-2} in the runtime. Alternatively, using the amplitude estimation subroutine, the ϵ -dependence could be reduced to $O(\epsilon^{-1})$.

The algorithm can be thought of as the reverse process of mapping a quantum circuit to a tensor network.

Dominant resource cost (gates/qubits)

The dominant cost of the algorithm is, on the one hand, the $\text{poly}(q^d)$ scaling, which can be substantial for highly connected graphs. More importantly though, for problems of interest is the value of Δ , which can grow exponentially with $|V|$ and require extremely high precision ϵ to give a meaningful answer. In other words, Δ sets the scale of the approximation.

The complexity of the quantum algorithm depends sensitively on the structure of the graph $G(V, E)$, on the tensors $\{M_v\}_{v \in V}$ and on the choice of the contraction pathway. A number of limiting cases are known [54]:

- There are tensor networks for which it is NP-hard to obtain a classical additive approximation of the full contraction, suggesting the classical hardness of the problem.
- There exist families of tensor networks for which the additive approximation in Eq. (24.1) is BQP-hard, suggesting that there exists a complexity separation between the classical and quantum problem.
- There are specific examples of tensor networks representing partition functions, for which the quantum approximation scale Δ is exponential in $|V|$, but with a smaller exponent than the best known classical additive approximation scheme. There exist other examples where the converse is true [54].

Furthermore, approximate contraction of a tensor network representing a quantum partition function of a positive semidefinite Hamiltonian has been shown to be complete for the one clean qubit (DQC1) model of quantum computation [290], which suggests that approximate contraction is likely classically hard, at least for certain specific instances. Classical algorithms for this problem that are not based on tensor networks have also been examined [565].

Caveats

The main caveat at present is that we do not have a good understanding of the structure of the network that allows for significant speedup on a quantum computer, due in part to the appearance of the complicated parameter Δ in the complexity statement. It is possible that the only situations where this is possible is when the tensor network can be mapped directly to a quantum circuit, without significant overhead. For example, in [616], a specific kind of tensor network called DMERA was shown to admit an exponential quantum speedup for approximate contraction because it arises from a specific kind of quantum circuit. The speedup is in the depth of the disentangling layer of the DMERA. A more critical caveat is that we do not understand when classical contraction algorithms are inefficient in practice. For instance, certain specific classes of tensor networks can be contracted efficiently [90, 1077].

Even quantum computational supremacy experiments [63], which were designed specifically to maximize the separation between quantum and classical simulation, allow for tractable tensor network simulations up to large system sizes (~ 50) and circuit depths (~ 30) [826], though these simulations become much more challenging if we allow for nonlocal gates. Another subtle point is that it is often, or even usually, not clear what magnitude to expect for the value of a contracted tensor network from just looking at tensor norms. For example, terms may have partially canceling phases. The additive error bound for the quantum algorithm might not be very helpful in such cases.

Finally, it is likely difficult to make a proper comparison between classical approximate methods (e.g., the corner transfer matrix) and the above quantum approximation schemes, as the classical and quantum approximation errors have very different origins, and the quantum algorithm cannot be simulated at scale. The quantum algorithm might thus be regarded as a new heuristic to be tested on a case to case basis once sufficiently powerful quantum hardware is available.

Example use cases

There is an obvious case where the quantum algorithm provides an advantage, and that is if you prepare a quantum circuit, and map it onto a tensor net-

work. Less trivial examples involve estimating partition functions of classical statistical mechanics models—although for this problem, good classical methods exist for the additive approximation [290]. Other applications involving large-scale tensor network contractions include condensed matter physics and molecular simulations. Inference problems [342] or differential equations simulation [449] might benefit from a quantum backend in some regimes, but a careful analysis has not yet been performed.

Further reading

- Pedagogical introductions to tensor networks [201, 150, 815].
- Quantum-inspired tensor network algorithms [602, 830, 390, 817].
- Complexity analysis of the quantum partition function problem [195].

