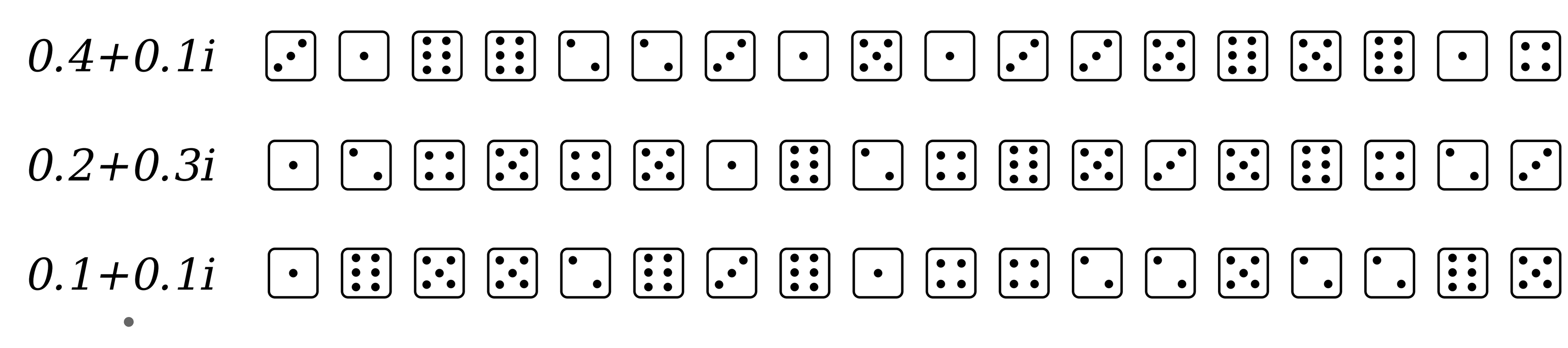
Tensor Networks

Weapons of Mass Simulation

Gregory Crosswhite
University of Washington

The Challenge

AN EPIC QUEST HAS BEEN UNDERTAKEN to build a quantum computer -- a new kind of computer founded upon the laws of quantum mechanics which could usher in a new era in computational science. Unfortunately, the same “quantum entanglement” (right) which gives such a computer its power is also a beast that makes large quantum systems difficult to simulate on a classical computer. On this poster, I present the technique of using tensor networks to slay this monster in many of its forms, as well as results I have obtained from applying these weapons to a realistic model.



QUANTUM ENTANGLEMENT IS THE presence of correlations in a quantum system. Consider a system of eighteen entangled quantum dice. Intuitively, one would like to roll each die separately and put the results together, but due to correlations between dice this will not give one complete information. Instead, one needs the complex-valued amplitudes (whose square gives the probability) of all 101,559,956,668,416 possible rolls, three of which are illustrated above.

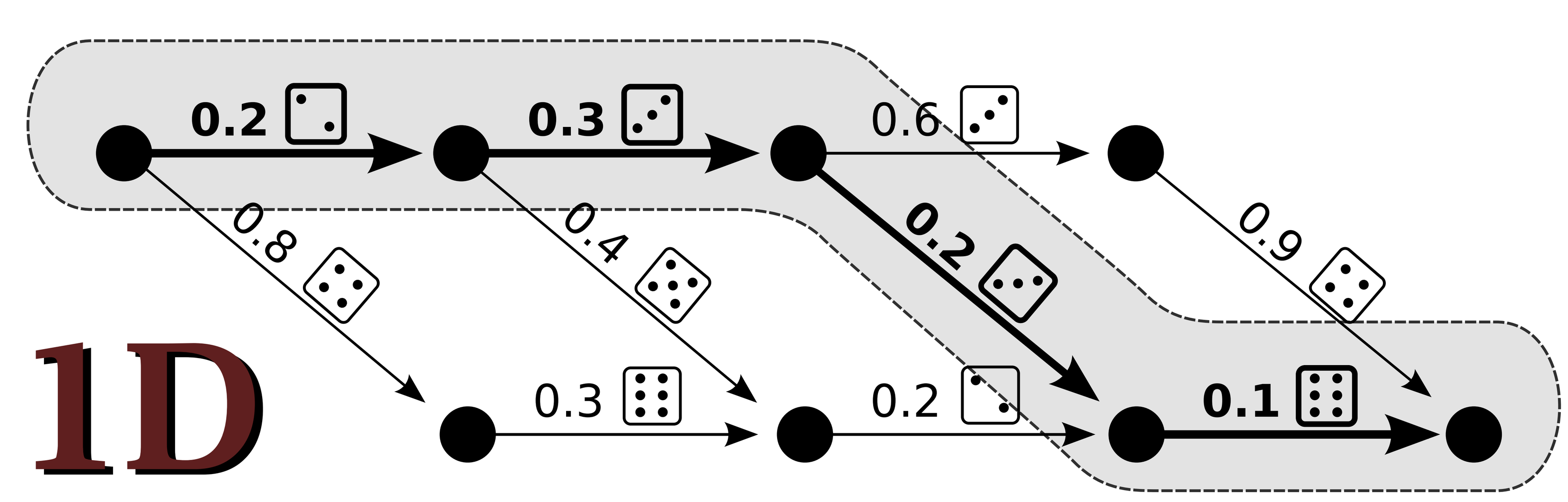
The Weapons

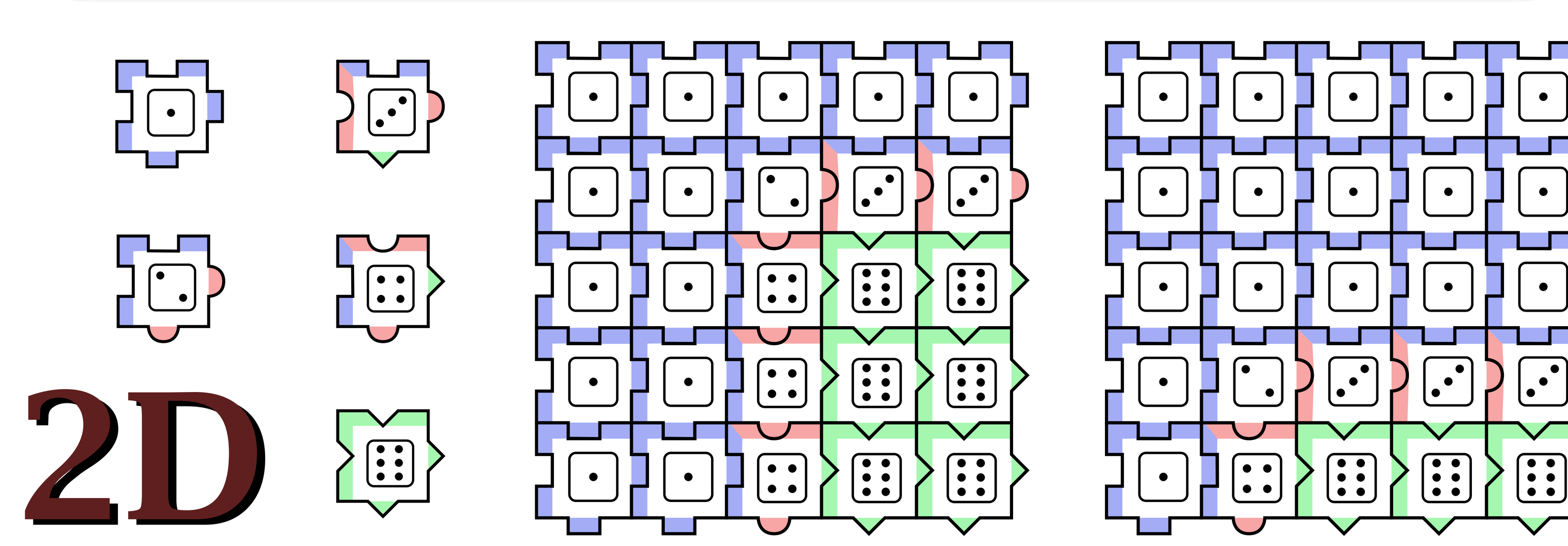
TENSOR NETWORKS ARE A WEAPON WE CAN wield to slay the beast of entanglement. Mathematically, they work by making the ansatz that the rank-N tensor describing a N-particle system of interest can be factored into a network of N smaller-rank tensors. For example, a rank-4 tensor representing the state of a one-dimensional 4-particle system of quantum dice could be factored into the inner product of four tensors of rank 2 or 3,

$$\psi^{\alpha\beta\gamma\delta} = \sum_{ijk} (S_1)_i^{\alpha} (S_2)_{ij}^{\beta} (S_3)_{jk}^{\gamma} (S_4)_k^{\delta}$$

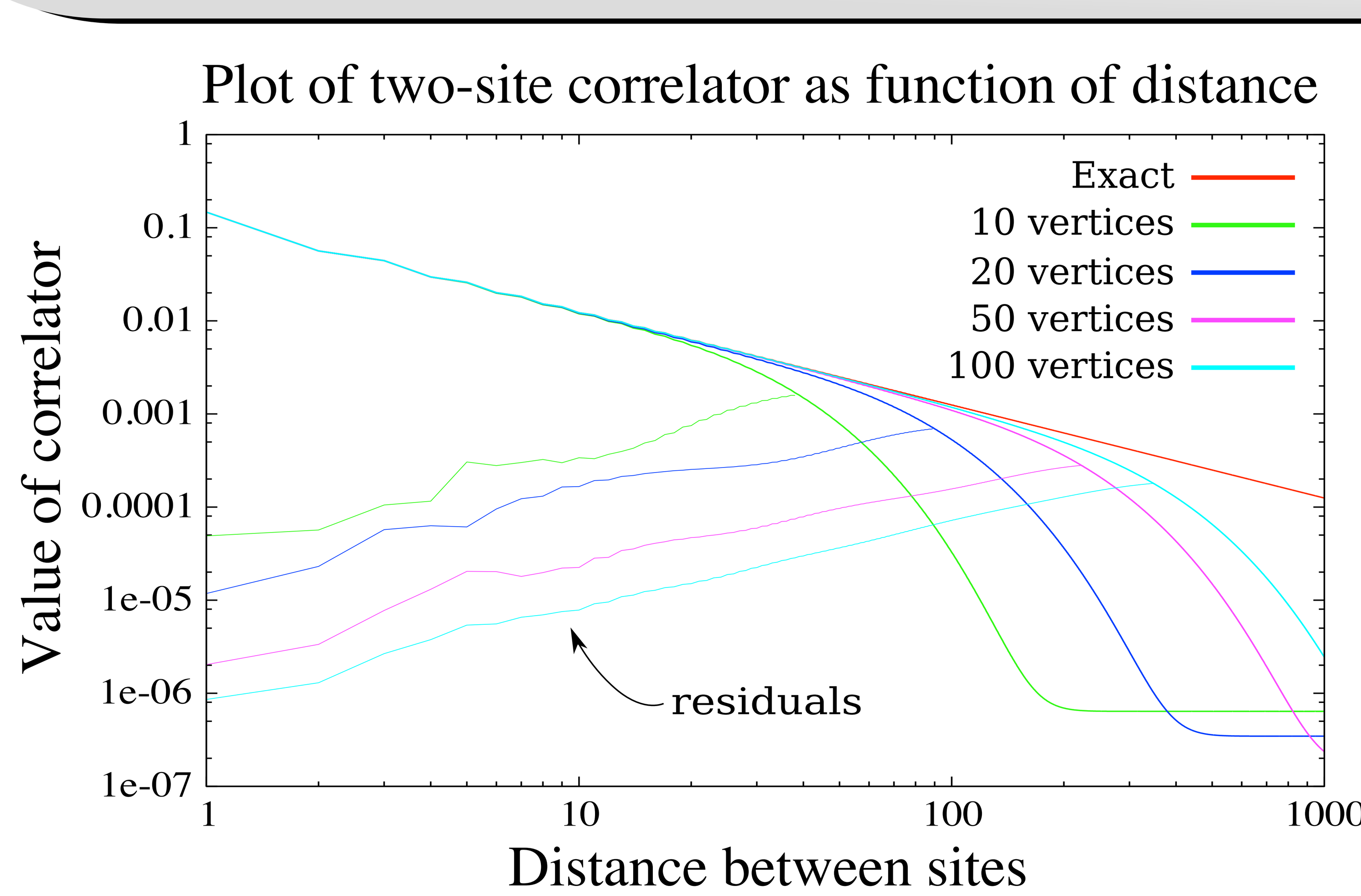
where the superscript indices represent physical observables (the sides of each of the dice) and the subscript indices denote signals that can be exchanged between dice. To visualize what is going on, consider the top picture on the right. In one dimension, one can construct a diagram as shown such that each possible roll of the dices corresponds to a walk through the diagram from left to right. The connecting vertices (corresponding to the subscript indices) allow the possible value of each die to depend on the values of the dice next to it. The amplitude of the roll is given by the product of the weights along the path. (Details in 1st ref.)

In two dimensions, a different sort of diagram is needed, but the idea is similar. Each die receives signals from the dice above and to their left, makes a decision, and then sends signals down and to the right; each possible decision that could be made is represented by a puzzle piece in the lower-right diagram. The result is that each possible roll of the grid of dice corresponds to a grid assembled from the given pieces. For example, for the particular choice of five pieces shown at right, at most one 2 can be rolled, although it can appear anywhere in the grid.





The Battle of Haldane-Shastry



ARMED WITH TENSOR NETWORKS, WE ATTACKED the Haldane-Shastry model, a one-dimensional spin lattice featuring long-range anti-ferromagnetic interactions. Since this model has been exactly solved, it provides a good test for our experimental techniques; furthermore, solution of this particular model has never before been attempted using tensor network techniques. We made the ansatz that the ground state of this model in the infinite limit could be represented in the form of the 1D diagram above where every site (tensor) is identical, and then used an optimization algorithm to solve for the wirings (site tensor elements). The two-point correlator (a measure of correlations between sites as a function of distance) is plotted on the left side. By increasing the number of “vertices” in our diagram, we were able to approximate the true correlator arbitrarily well, demonstrating that our algorithm works. (Details in 2nd ref.)

Scrolls of Reference: (in print)

G. M. Crosswhite & D. Bacon, Finite automata for caching in matrix product algorithms, arXiv:0708.1221

G. M. Crosswhite, A. C. Doherty & G. Vidal, Applying matrix product operators to model systems with long-range interactions, arXiv:0804.2504