Efficient construction of robust graphical architectures

Nathan X. Kodama, Vanessa Job, and Laura M. Monroe
Los Alamos National Laboratory
Los Alamos, NM, 87501 USA

July 7, 2022

Abstract

We build on recent protocols for training sparse Boltzmann machines by exploiting symmetries in Boolean multiplication. Our approach leverages factor graph correspondences and efficient message-passing algorithms for training large, complex architectures across classical and quantum hardware. We discuss the prospect of combining symmetries and graphical correspondences to approximate Boolean functions and evaluate their expressive power.

Contents

1 Overview 2

2 Symmetries of Boolean multiplication 2

3 Expressive variational ansätze via factor graphs 4

4 Enhanced representation learning in deep networks 6

5 Discussion and next steps 8

A Sparse encoding of Boolean multiplication 12

1 Overview

We recently used the solution of a constraint satisfaction problem (CSP) to initialize the weights of a sparse, semi-restricted Boltzmann machine (SS-RBM). This protocol increased the robustness of a 3-bit adder over the baseline performance of the CSP solution alone; moreover, we showed that the initialization was *necessary* for the convergence of the SS-RBM to a good solution.[1]

Here we explore the efficient construction of graphical architectures for Boolean circuits. Our approach builds on our protocol above by (*i*) making the invariant components of n-bit multiplication more robust via sparse graphs and (*ii*) exploiting symmetries in factor graph correspondences to derive efficient training algorithms that can be employed across classical and quantum hardware.

2 Symmetries of Boolean multiplication

Permutation-invariant and shift-equivariant components. The n-bit multiplication circuit comprises layers of ripple-carry adder (RCA) circuits, whose inputs are given by a sub-layer of AND gates. Both of these components exhibit symmetries. The AND gate has an output that is permutation-invariant with respect to the inputs, while the RCA circuit has an output that is shift-equivariant up to its boundary conditions. Therefore, the invariant structures of the Boolean multiplication circuit are the AND gate and Full-Adder (FA). We may exploit these symmetries in the inductive bias of our learning algorithm, i.e. a *geometric prior* [4,5]. In the case of Boolean multiplication, an efficient learning algorithm reduces the high-dimensional problem of learning n-bit multiplication to the low-dimensional problem of learning simple Boolean function mappings. This approach points to a local (greedy) learning algorithm that the robustness of the architecture.[2]

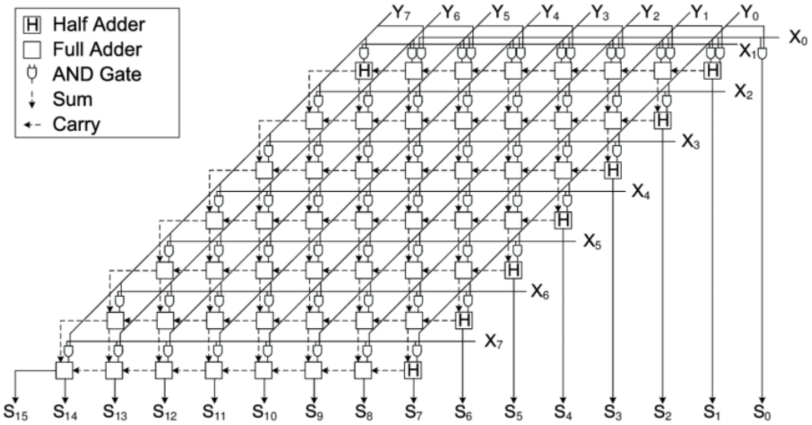

Figure 1: Symmetries and invariants of the Boolean multiplication circuit [3].

[1] Our protocol is reminiscent of unsupervised pre-training: a well-known technique for overcoming local optima in deep belief networks [1,2].

[2] Several other reasons for leveraging invariant components and thereby minimizing the number of parameters is to increase the speed of convergence and generalizability [6].

Reversible and scale-invariant circuits. The reversibility of quantum-assisted Boolean multiplication implies that we can factorize an integer N with a b-bit representation and $N \leq 2^b$ [7]. This is limited by the size of the quantum processing unit (QPU)—b=16 for the D-Wave 2000Q—and noisy control and measurement parameters. However, we can leverage the scale-invariance of Boolean multiplication to solve larger problems. We propose tiling many small multiplication circuits to construct a large multiplication circuit. By making multiple calls to the QPU, we achieve efficient scaling: the number of sequential calls scales as $O(b^2)$ and the number of parallel calls scales as $O(b)$. The best-known classical factoring algorithms scale as $O(2^{\alpha b})$ for $0 \leq \alpha \leq 1$; thus, factoring with the QPU offers an exponential speedup.

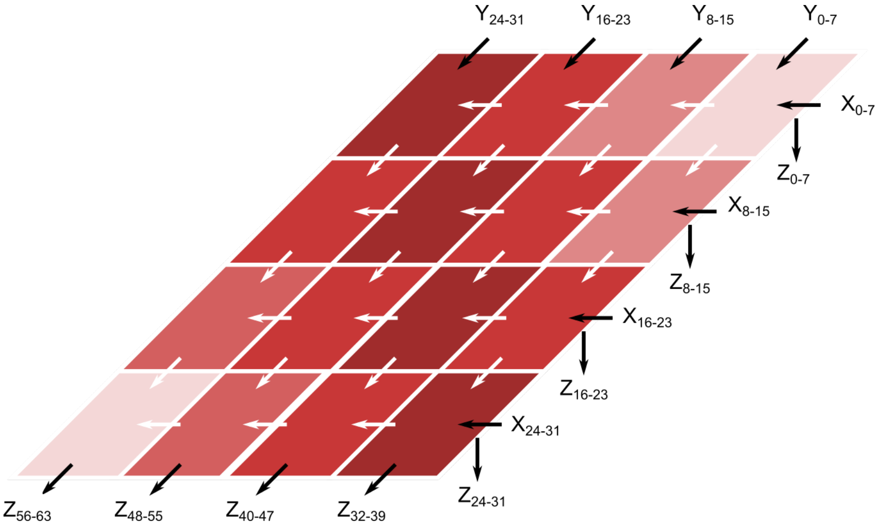

Figure 2: Scale-invariant structure of reversible multiplication (factoring) and construction via tiling.

3 Expressive variational ansätze via factor graphs

Our construction of the multiplication circuit maps to a class of tensor networks known as projected entanglement pair states (PEPS), which cannot be efficiently simulated on classical hardware [8]. The PEPS network is an induced minor of the chimera graph, i.e., it is obtained by (i) contracting edges in the $K_{4,4}$ complete bipartite graph of each unit cell to form a K_5 complete graph, (ii) representing the complete graph and chimera graph as factor graphs, and (iii) summing over shared indices to obtain the PEPS decomposition (Fig. 3). We propose using PEPS as an expressive variational ansätz that can be embedded on the chimera graph and efficiently sampled from the D-Wave.

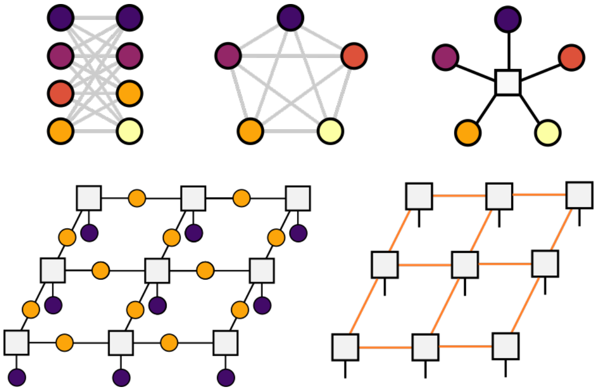

Figure 3: Mapping the chimera graph to a PEPS tensor network via factor graphs.

The expressive power of PEPS is greater than other mappings of the chimera graph, such as *bipartite* and *chains* (Fig. 4). The bipartite mapping is a sparse restricted Boltzmann machine, which can be efficiently simulated; this structure performs poorly on image classification tasks [9] and slightly worse than chains, since the latter is harder to simulate classically [10]. PEPS derives its expressive power from its large latent space, which is 1.75× larger than the other mappings. While these other maps use half of the QPU's resources (1024 qubits), PEPS can devote all but one qubit per unit cell to the latent space (1792 qubits). Utilizing large latent spaces is a necessary condition for quantum advantage since this is a regime in which classical sampling approaches become prohibitively expensive due to slowly mixing models [10].[3]

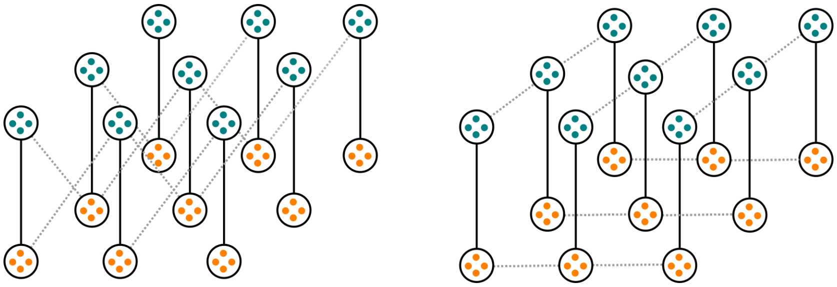

Figure 4: Bipartite and chains mappings of the chimera graph.

[3] A sufficient condition for quantum advantage is the presence of complex, multimodal distributions.

Here we leverage the expressive power of PEPS in 2D problems such as Boolean multiplication and image processing. In Boolean multiplication, the exact solution exists in the hypothesis space of PEPS (Fig. 5). This variational ansätz can be initialized *locally* using the symmetries and invariants described in Section 1, before being optimized *globally* via a learning algorithm, such as contrastive divergence [11] or quantum tempering [12] according to clamped inputs/outputs and constrained latents. Note that we have extracted the action of AND gates to a linear pre-processing step (an outer product, $O(n^2)$) to maximize the throughput of the quantum nonlinearity enacted by the QPU. Therefore, reversing the operation of the protocol for factoring requires a linear post-processing step (diagonalization, $O(n^3)$) to recover the Boolean input strings.

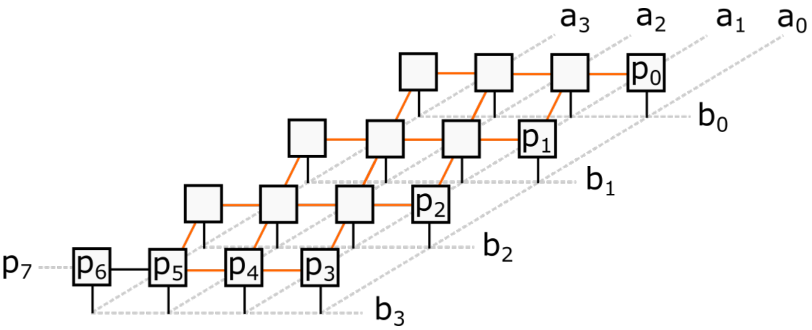

Figure 5: Application of the PEPS variational ansätz to multiplication.

In Section 2, we proposed an initialization that exploits symmetries in the circuit; in this section, we have shown how PEPS can maximize the size of the latent space. During the training, we may move between these extremes by enforcing local constraints: we may additionally clamp a set of intermediate results to visible units, which reduces the size of the latent space by enforcing a stronger inductive bias. Conversely, relaxing these constraints could allow the learning procedure more flexibility to find more well-separated basins of attraction (higher capacity and robustness). The set inputs/outputs shown in (Fig. 5) is irreducible, but the number of additional visible units is a hyperparameter that should be tuned with cross-validation.

4 Enhanced representation learning in deep networks

Our tiling approach to scaling Boolean multiplication/integer factorization induces a deep network design that cascades information in 2D and aggregates results along the boundaries of a grid. Deep networks offer state-of-the-art performance on image processing tasks such as classification, object detection, and segmentation. We focus on deep network designs for *image classification* whose quantum latent spaces cannot be efficiently trained on classical hardware. Our approach adheres to a *linear baseline rule*, which excludes the use of nonlinear transformations, including those carried out by neural networks, from pre- and post-processing steps. Such a provision is necessary because generic nonlinearities can add expressive power to the overall algorithm *in spite of* of the transformation made by the QPU.

The most straightforward experiment that combines the linear baseline rule with a quantum latent space is shown in (Fig. 6), which depicts the discrimination of a handwritten digit from the space of possible images (3 vs. all). A linear feature map projects a local set of pixels onto a parameter of the input qubit (i.e., the bias current); then, a PEPS network may be optimized to either generate handwritten digits (unsupervised) or lower/raise the energy of those inputs corresponding to 3 (supervised). The result is a nonlinear kernel mapping from the original feature space, which may not be linearly separable, to a quantum-enhanced feature space that may be linearly separable by the decision surface of a linear SVM.

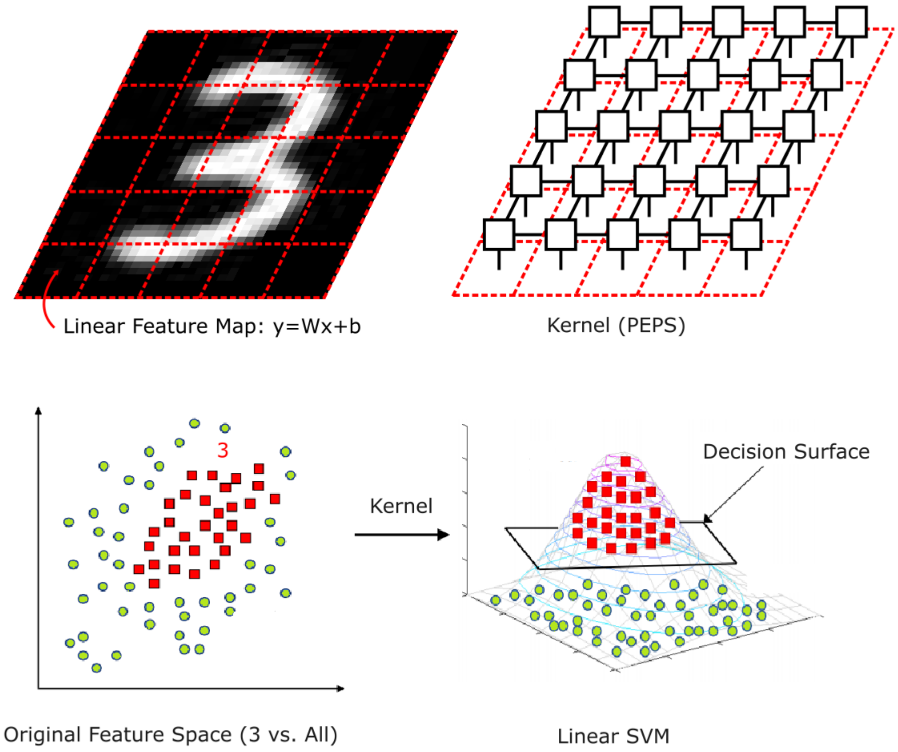
Linear Feature Map: y=Wx+b
Kernel (PEPS)
3
Kernel
Decision Surface
Original Feature Space (3 vs. All)
Linear SVM

Figure 6: A direct application of PEPS to image classification. A linear feature map over a block of pixels in an image of a 3 is nonlinearly encoded in each open index of PEPS. The network acts as a variational kernel transformation from an original feature space, which is not linearly separable, to an enhanced feature space that may be linearly separated (SVM image credit: Apurv Jain).

Identifying symmetries benefits both the speed of convergence and the generalizability of the network. Fortunately, the class label of an image is *shift-invariant*: if we translate the image on the 2D grid, its class label remains the same. This implies that weight sharing techniques may be enforced across the grid to improve the convergence and generalizability of the network. However, there is a risk of overly reducing expressive power when we decrease the number of parameters. Tuning the amount of weight sharing via local receptive fields would modulate tradeoffs between trainability and expressivity. Still, factors whose receptive fields transform locations of the image with small effects on the overall task, such as the image boundaries, will likely have a variance that goes to zero, i.e., those hidden variables will become inactive, thereby reducing the size of the overall latent space.

We may circumvent this problem by constructing a *delocalized* linear feature encoding. Instead of having a local feature encoding over a grid of pixels, we can implement a distributed feature encoding across the image according to some regular structure [9]. If the feature maps are random linear projections across all inputs (Fig. 7, left), we have theoretical guarantees on rates of convergence: specifically, random Fourier bases may be used to obtain good function approximations with error that scales as $O(1/\sqrt{K} + 1/\sqrt{N})$ for K basis projections and N training examples [13–15].

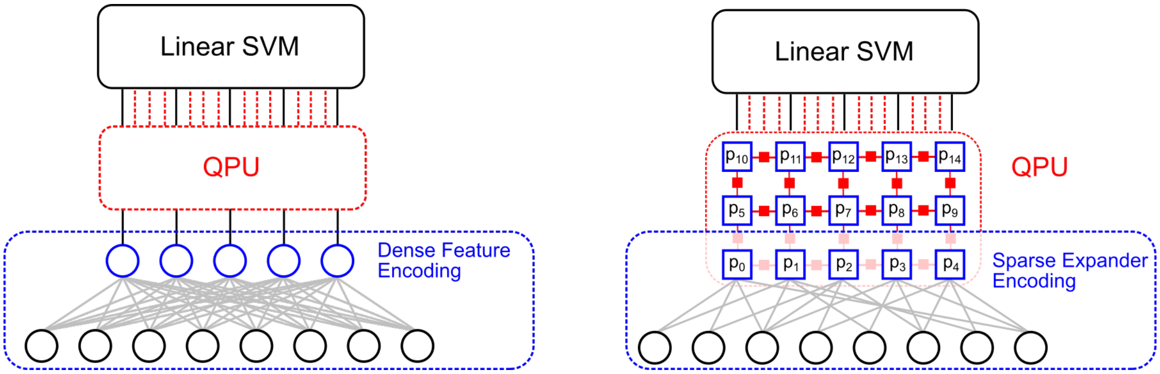

Figure 7: Machine learning pipeline with distributed feature encodings for dense, fully connected layers over continuous data (left) and sparse expander codes over binary data (right). For illustrative clarity, the sparse expander encoding is restricted to a single row of the QPU—in reality, inputs may be distributed across the entire QPU.

We may implement an even stronger inductive bias that directly encodes features on the QPU using the parity-check structure of a strong error-correcting code (Fig. 7, right). In this case, we sparsely distribute features across the QPU by initializing each factor tensor as a localized parity-check. We are interested in inducing the properties of sparse expander graphs [16]—namely, exponential capacity, robustness, and self-decoding. With these theoretical guarantees, we would have control over the design of exponential storage capacity and robustness of the network. We derive a protocol for sparsely and robustly encoding Boolean multiplication on the Chimera graph that leads to a $\sim 3\times$ increase in the allowable problem size, from 8-bit to 25-bit multiplication (Appendix 1) with guaranteed robustness—i.e., the number of correctable errors $N_{errors} \geq 6.9$). Further, we generalize to the Pegasus graph, yielding a sparse encoding of 152-bit multiplication with $N_{errors} \geq 36.7$.

5 Discussion and next steps

Since the sparse expander network implements a strong error-correcting code, it can encode arbitrary states with exponential capacity and thus risk overfitting traditional machine learning problems. In these cases, we explore cross-validation over the parameters of the expander graph. Further, we may relax the local parity constraints on the QPU during a variational search (e.g., quantum tempering) [12].

Building on the observations here, we describe a roadmap towards robust graphical architectures. Our general approach is to compare informative inductive biases by initializing our architectures with (i) a geometric prior that exploits the symmetries and invariants of Boolean multiplications and (ii) a sparse expander code that has exponential capacity, robustness, and self-decoding properties.

Using this framework, we explore a hierarchy of local learning problems and assess the robustness at various levels, from gates to circuits; we can evaluate the added robustness of a global variational learning algorithm. We may continue to exploit the symmetries and invariant structures during inference when we utilize efficient message-passing algorithms over the factor graph representation of the variational ansätz. Combining these architectures with symmetry, sparsity, and connectivity constraints, we derive a novel architecture that admits an efficient message-passing algorithm for classical and quantum hardware.

Roadmap:

- Evaluate the resilience of RCA+ECC techniques and robust Boltzmann machines:
  - Mitigate errors at higher noise (BER > 1) with LDPC codes, while varying densities and local constraints
  - Learn errors at lower and higher noise by replacing RCA+ECC with Boltzmann machines, utilizing effective temperature estimation and error-corrected sampling
  - Combine strong ECC and Boltzmann machines via bipartite expander codes
- Replace multilayer circuits, such as the BWM circuit, with deep Boltzmann machines:
  - Replace the BWM with layers of RBMs, where the BWM is a minor embedding of the deep Boltzmann machine
  - Study wide and deep Boltzmann machines embedded on the chimera graph for mapping circuit inputs to outputs
  - Jointly train an over-complete VAE in a balanced and symmetric configuration on the chimera graph
- Explore theoretical connections and novelties between ML and ECC within the framework of probabilistic graphical models
  - Probe the connection through message-passing algorithms
  - Compare the parity-check formalism with binary polynomials obtained through layers of arithmetic circuits

Acceptance criteria:

- Compare regular and algorithmic graphical embeddings on the chimera graph of the D-Wave throughout and characterize the resulting optimization and sampling statistics
- Credibility intervals, given reasonable runtimes on HPC resources
- Reproducible code, shared in the Git repository

References

[1] G. E. Hinton, S. Osindero, and Y. W. Teh, “A Fast Learning Algorithm for Deep Belief Nets,” *Neural Computation*, 2006.

[2] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” *Advances in Neural Information Processing Systems*, no. 1, pp. 153–160, 2007.

[3] M. Själander, “Efficient Reconfigurable Multipliers Based on the Twin-Precision Technique,” 2006.

[4] M. M. Bronstein, J. Bruna, Y. Lecun, A. Szlam, and P. Vandergheynst, “Geometric Deep Learning: Going beyond Euclidean data,” *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[5] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, “Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges,” 2021.

[6] Y. LeCun, “Generalization and Network Design Strategies,” in *Connectionism in Perspective* (R. Pfeifer, Z. Schreter, F. Fogelman, and L. Steels, eds.), (Zurich, Switzerland), Elsevier, 1989.

[7] E. Andriyash, Z. Bian, F. Chudak, M. Drew-brook, A. D. King, W. G. Macready, and A. Roy, “Boosting integer factoring performance via quantum annealing offsets,” 2016.

[8] R. Orús, “Tensor networks for complex quantum systems,” *Nature Reviews Physics*, vol. 1, no. 9, pp. 538–550, 2019.

[9] V. Dumoulin, I. J. Goodfellow, A. Courville, and Y. Bengio, “On the Challenges of Physical Implementations of RBMs,” in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI’14, pp. 1199–1205, AAAI Press, 2014.

[10] W. Winci, L. Buffoni, H. Sadeghi, A. Khoshaman, E. Andriyash, and M. H. Amin, “A path towards quantum advantage in training deep generative models with quantum annealers,” *Machine Learning: Science and Technology*, vol. 1, p. 45028, nov 2020.

[11] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” *Neural Computation*, 2002.

[12] N. Chancellor, “Modernizing quantum annealing using local searches,” *New Journal of Physics*, vol. 19, feb 2017.

[13] A. Rahimi and B. Recht, “Uniform approximation of functions with random bases,” in *46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 555–561, 2008.

[14] A. Rahimi and B. Recht, “Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning,” in *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference*, vol. 1, pp. 1313–1320, 2009.

[15] A. Rahimi and B. Recht, “Random features for large-scale kernel machines,” in *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference*, no. 1, pp. 1–8, 2009.

[16] R. Chaudhuri and I. Fiete, “Bipartite expander Hopfield networks as self-decoding high-capacity error correcting codes,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

A Sparse encoding of Boolean multiplication

In n-bit multiplication, the number of valid configurations grows exponentially as 2^{2n} an the total number of configurations grows exponentially as 2^{4n}. Direct implementations of basic network architecture such as fully-connected Hopfield networks and restricted Boltzmann machines are guaranteed to fail since the number of possible coding states in these networks grows *linearly* with the network size.

A bipartite expander network is derived from a class of low-density parity-check codes with strong connectivity. Under certain parity-check constraints and network design structures, the networks exhibit several important properties for hardware embeddings of Boolean multiplication: exponential capacity, robustness, and self-decoding. The key idea is to build higher-order constraints C_i, which can be expressed as an Ising Hamiltonian, $E(s) = -\sum_{C_i} \Pi_{i \in C_i} s_i$. Referring to the inset of Fig. 7 (right), these higher-order parity constraints are mapped to a bipartite graph of constraint nodes (blue) and N visible nodes (black). If the average degree of visible nodes satisfies $\langle z \rangle \geq 5$ and the average degree of constraint nodes satisfies $\langle z_C \rangle \geq C$, then a randomly connected bipartite graph is an expander graph with probability $1 - O(1/N)$.

If the ratio of average degrees of visible and constraint nodes $\hat{z} = \langle z \rangle / \langle z_c \rangle$, satisfies