

Beyond Belief Propagation: Cluster-Corrected Tensor Network Contraction with Exponential Convergence

Siddhant Midha^{1,*} and Yifan F. Zhang^{2,*}

¹*Princeton Quantum Initiative, Princeton University, Princeton, New Jersey 08544, USA*

²*Department of Electrical and Computer Engineering, Princeton University, Princeton, New Jersey 08544, USA*



(Received 24 October 2025; accepted 28 May 2026; published 9 July 2026)

Tensor network contraction on arbitrary graphs is a fundamental computational challenge with applications ranging from quantum simulation to error correction. While belief propagation (BP) provides a powerful approximation algorithm for this task, its accuracy limitations are poorly understood and systematic improvements remain elusive. Here, we develop a rigorous theoretical framework for BP in tensor networks, leveraging insights from statistical mechanics to devise a *cluster expansion* that systematically improves the BP approximation. We prove that the cluster expansion converges exponentially fast if an object called the *loop contribution* decays sufficiently fast with the loop size, giving a rigorous error bound on BP. We also provide a simple and efficient algorithm to compute the cluster expansion to arbitrary order. We demonstrate the efficacy of our method on the two-dimensional Ising model, where we find that our method significantly improves upon BP and existing corrective algorithms such as loop series expansion. Our work opens the door to a systematic theory of BP for tensor networks and its applications in decoding quantum error-correcting codes and simulating quantum systems.

DOI: [10.1103/sdlx-d3tn](https://doi.org/10.1103/sdlx-d3tn)

I. INTRODUCTION

Tensor networks comprise a set of powerful mathematical and computational tools widely used in condensed matter physics and quantum information science. Originally developed for quantum many-body systems [1,2], tensor networks have evolved into a unifying framework that connects diverse areas of physics [3–7] and computer science [8–12].

Despite their conceptual elegance and broad applicability, the practical utility of tensor networks is fundamentally limited by the computational complexity of tensor contraction. Contracting a tensor network—summing over all internal indices to compute the final result—is central to extracting physical quantities from the network representation. However, this operation generally requires exponential runtime in the system size [8,13,14], motivating the development of polynomial-time algorithms for approximate network contraction.

Many such algorithms have been developed, often exploiting special structures in the tensor network to simplify contractions. For example, time evolving block decimation (TEBD) [15,16] leverages the bounded growth of entanglement to truncate the bond dimension. Other algorithms exploit the network geometry; in particular, for geometries without loops (such as one-dimensional matrix product states [17,18] and tree tensor networks [19]), exact contraction becomes efficient if performed in the right order.

Belief propagation (BP) [20–24], a classical algorithm rooted in computer science and statistical physics, has recently emerged as a promising candidate for approximate contractions of tensor networks [25,26]. Originally developed by Pearl for probabilistic inference in graphical models [20], BP found its theoretical foundation in Bethe’s work on lattice statistical mechanics [21]. BP has since proven useful in tasks such as decoding classical and quantum low-density parity check (LDPC) codes [27–33], machine learning [34,35], and optimization [36,37]. While BP was developed in the context of classical probability theory, it is equally applicable to quantum systems and has been widely adopted [38–47]. Notably, BP-based techniques have been used to classically simulate major quantum experiments, challenging claims of quantum advantage [48,49].

BP offers several advantages over other methods: it is polynomial time and parallelizable, applies to arbitrary

*These authors contributed equally to this work.

[†]Contact author: sm7456@princeton.edu

[‡]Contact author: yz4281@princeton.edu

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article’s title, journal citation, and DOI.

geometries, and becomes exact on networks without loops. However, the conditions under which BP is a good approximation are still not clear, especially in loopy geometries. In addition, unlike methods such as TEBD, which can be controllably improved (e.g., by increasing the bond dimension), the traditional BP method is inherently rigid and lacks a tuning parameter to trade off computational resources for lower error rates.

In this work, we develop a systematic theory of belief propagation (BP) for tensor network contractions to address these challenges. Our main contributions are: (i) establishing rigorous control over the difference between the exact value Z and the BP contraction Z_0 , and (ii) designing an efficient algorithm that systematically corrects the BP error with exponential convergence.

We achieve our results through a novel approach that overcomes fundamental limitations of existing methods. In Refs. [50,51], the authors take inspiration from earlier work in statistical inference [52–59] to construct a Taylor series known as the *loop expansion* in order to correct the BP approximation value toward the ground truth. The naïve loop expansion starts from the BP value Z_0 at zeroth order and sums corrections called *loop tensors* that grow larger at higher orders. While this seems like a natural tuning knob—when fully summed, the loop expansion yields Z —the method suffers from a critical flaw: it does not generally converge, even when individual loop tensors decay exponentially. The fundamental problem is that the expansion includes disconnected loops whose number grows combinatorially with size, overwhelming any exponential decay and causing divergence. The work [50] made the first step in mitigating the combinatorial growth through multiple self-consistent methods.

To resolve this convergence failure, we leverage insights from statistical mechanics to construct an entirely different series: a *cluster expansion* that converges to the logarithm of the ground truth, $\log(Z)$, rather than Z itself. Starting from $\log(Z_0)$, our method sums modified corrections called *clusters*, derived from loop tensors. Crucially, only *connected clusters* contribute to our expansion, and their number grows at most exponentially—not combinatorially. This fundamental difference ensures that our cluster expansion converges exponentially fast, solving the convergence problem that plagues the loop series method.

We rigorously prove the exponential convergence of our cluster expansion, provided that loop tensors decay exponentially with a sufficiently large exponent. This resolves the two main challenges of BP: it characterizes when BP is accurate and provides a polynomial-time algorithm to systematically improve it—something that loop series expansions cannot achieve due to their inherent convergence problems.

To understand when loops decay and the cluster expansion converges, we conduct extensive numerical experiments computing the free energy density of the 2D Ising

model and benchmark it against the exact solution. We observe distinctive behaviors across the phase transition: while BP performs well deep in the high-temperature and low-temperature phases, it deviates from the ground truth because of the presence of long-ranged fluctuations at the critical point. There, we show that cluster expansion significantly improves the BP error and converges faster than loop expansion, independent of the system size.

The change from the loop expansion to the cluster expansion is physically intuitive. Borrowing wisdom from statistical mechanics, we interpret tensor networks as partition functions. As an object that changes multiplicatively with system sizes, it is inherently unstable and is hard to approximate with a series expansion. In contrast, the logarithm of the partition function, or the free energy, is an extensive quantity that changes additively with system sizes. It has an additive response to local perturbation and can thus be approximated with a series expansion. This intuition is rigorized in our convergence proof and is supported by our numerical experiments.

A. Comparison with existing and concurrent work

We compare our results with earlier and concurrent works. References [52,53] first introduce the loop series expansion in probabilistic graphical models. Reference [50] generalizes the loop series expansion to tensor networks. Notably, they propose several methods to overcome the divergence of the naïve loop expansion. Their key intuition is the extensivity of free energy which also motivates our paper. In particular, the combinatorial toolbox we will use formalizes the counting argument underlying their “single-excitation” and “multi-excitation” approximation. Our results advance this line of work in three key respects. First, the cluster expansion yields explicit expressions for correction terms at all orders. Second, it applies beyond the thermodynamic limit, whereas the previous methods are restricted to that setting. Lastly, we resolve an open question by showing that sufficiently fast loop decay guarantees exponential convergence of the cluster expansion.

There are also earlier attempts to derive modified series expansion with better convergence than the naïve loop expansion. Notable examples include the *linked cluster expansion* [60], also known as the *cluster-cumulant expansion* [61] in the statistical inference literature. Reference [51] employs the linked cluster expansion as one of the subroutines to compute observables in time-evolved quantum systems. Concurrent to our work, Ref. [62] appeared which employs the linked cluster expansion to perform extensive numerical simulations of quantum ground states. We note that while the linked cluster expansion is conceptually very similar to the cluster expansion derived in our work, they are formally two different series. See Sec. III C for an example. However, we believe that our theory of convergence should apply to

linked cluster expansion as well, since both series reflect fundamentally the same physics.

B. Organization

This paper is organized as follows. In Sec. II, we review the belief propagation algorithm and its connection to tensor networks. In Sec. III, we introduce the loop and cluster expansions, present our main convergence theorem, and discuss its implications. In Sec. IV, we outline the algorithmic procedure for computing the cluster expansion. In Sec. V, we present numerical results on the two-dimensional Ising model where the exact solution is known. Finally, in Sec. VI, we discuss potential extensions and applications of our work.

II. BELIEF PROPAGATION

In this section, we define our notation and introduce the BP algorithm. We then introduce the loops series expansion and show why it does not converge in general.

A. Tensor networks

We consider a tensor network, with no open indices, defined on a graph $G = (V, E)$ on N sites with sets of vertices V and edges E . For each vertex $v \in V$, we denote the neighbors of v in G as $\mathcal{N}(v)$. The degree of a vertex in the graph is denoted $d(v) := |\mathcal{N}(v)|$. The degree of a graph is denoted $\Delta(G) := \max_{v \in V} d(v)$. Next, we define the notion of a tensor network on the graph. For edge $(v, w) \in E$ we associate the *bond* Hilbert space $\mathcal{B}_{vw}$ with $\dim(\mathcal{B}_{vw})$ the *bond dimension* on that edge, which we take to be uniform and equal to χ without loss of generality. Each vertex $v \in V$ is then equipped with a tensor $T_v \in \bigotimes_{n \in \mathcal{N}(v)} \mathcal{B}_{nv}$.

We refer to the triple $\mathcal{T} = (\{T_v\}_{v \in V}, V, E)$ as a *tensor network*. These networks have no “open” indices, and will be used for the belief propagation algorithm, as detailed in the following. Defined this way, the contraction of all the tensors in the network is a scalar,

$$\mathcal{Z}(\mathcal{T}) = \star_{v \in V} T_v, \quad (1)$$

where $\star$ denotes contraction of tensor indices. Following the convention in statistical mechanics, $\mathcal{Z}(\mathcal{T})$ is a sum of local terms over local Hilbert spaces, which is what usually defines a partition function in statistical mechanics. In fact, any partition function of a local statistical mechanical model can be written as a tensor network inheriting the locality of the model. Generalizing this language, we refer to $\mathcal{Z}(\mathcal{T})$ the *partition function* of the tensor network. Note that $\mathcal{Z}(\mathcal{T})$ is strictly more general than the partition function in statistical mechanics: in general, tensors T_v can be complex, while in statistical mechanics we only sum over non-negative quantities.

Following the same intuition, we will define the *free energy* as the logarithm of the partition function, generalizing the definition in statistical mechanics.

$$\mathcal{F}(\mathcal{T}) = \log \mathcal{Z}. \quad (2)$$

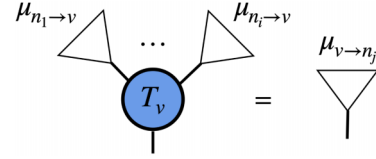
One might worry that $\mathcal{Z}$ can be complex and thus rendering $\mathcal{F}(\mathcal{T})$ multi-valued. In the following discussion, we will always choose a normalization such that $\mathcal{Z}$ is positive and store the phase information separately. This ensures the uniqueness of $\mathcal{F}(\mathcal{T})$ in the neighborhood where cluster expansion operates.

B. Belief propagation

Now we introduce the belief-propagation procedure. It begins with defining *message tensors* on each edge of the graph, with $\mu_{v \rightarrow w} \in \mathcal{B}_{vw}$ denoting the message from node v to w . We define a set of fixed-point messages on the network through the notion of *self-consistency* between the tensors and the messages. This requires that the contraction of all but one incoming message on any vertex $v \in V$ must result in the outgoing message on that edge. Mathematically, the self-consistent set $\mathcal{M} = \{\mu_{v \rightarrow w}\}_{v, w}$ satisfies for each $v \in V$ and each $s \in \mathcal{N}(v)$,

$$\left(\bigotimes_{n_i \in \mathcal{N}(v)/\{n_j\}} \mu_{n_i \rightarrow v} \right) \star T_v = \mu_{v \rightarrow n_j}. \quad (3)$$

Schematically,



where rank-one $\triangleright$ denotes message tensors and T_v is the local tensor at vertex v . In practice, the message tensors are obtained through iteration of the self-consistency equation until convergence [see Sec. IV B for details].

To evaluate the BP approximation, we first compute $I_{vw} = \mu_{v \rightarrow w} \star \mu_{w \rightarrow v}$, that is the inner product between $\mu_{v \rightarrow w}$ and $\mu_{w \rightarrow v}$ [63]. For each $v \in V$, the local contribution to the partition function is then given as,

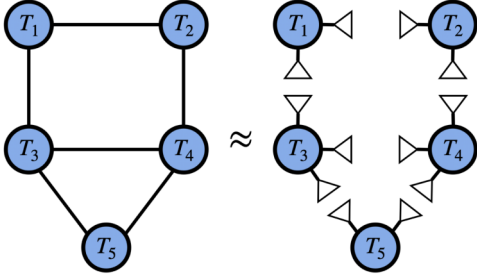
$$Z^{(v)} := \left[\bigotimes_{n \in \mathcal{N}(v)} \frac{\mu_{n \rightarrow v}}{\sqrt{I_{nv}}} \right] \star T_v, \quad (4)$$

where we take the principal branch of the square root when I_{vw} is complex. The BP vacuum solution to the partition function and the free energy is then given as,

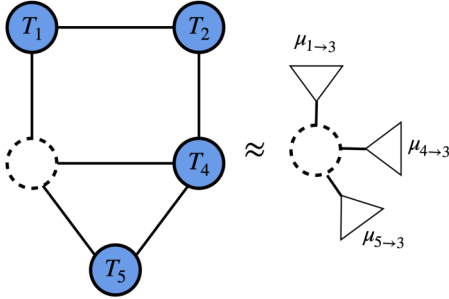
$$Z_0 = \prod_{v \in V} Z^{(v)}, \quad (5)$$

$$F_0 = \sum_{v \in V} \log Z^{(v)}. \quad (6)$$

Here, Z_0 forms the BP approximation to $\mathcal{Z}$. This is graphically depicted as the following,



Viewed as a mean-field approximation, the “traditional” BP algorithm is used to approximate the exact partition function with the vacuum solution Z_0 . Physically, the message tensors can be viewed as a rank-one approximation of the influence of a complex environment. The local reduced density matrices are then given by contracting the rank-one environments into the local tensors as follows,



C. Loop series expansion

The *loop series expansion* [50] for the self-consistent messages in terms of the “generalized loops” on the graph can be written as follows. For each edge $e = (r, s) \in E$, consider the identity $\mathbb{1}$. We define the orthogonal projector $\mathcal{P}_{rs}^\perp$ by expanding the identity as,

$$\mathbb{1} = \frac{|\mu_{r \rightarrow s}\rangle\langle\mu_{s \rightarrow r}|}{I_{rs}} + \mathcal{P}_{rs}^\perp. \quad (7)$$

This ensures that $\mathcal{P}_{rs}^\perp|\mu_{r \rightarrow s}\rangle = 0$ and $\langle\mu_{s \rightarrow r}|\mathcal{P}_{rs}^\perp = 0$, hence $\mathcal{P}_{rs}^\perp$ carries contributions orthogonal to the BP vacuum.

Now, consider the problem of evaluating the partition function $\mathcal{Z}$. Inserting the identity above at each edge in the network, one obtains $2^{|E|}$ terms. Each term can be expressed by an $|E|$ -bit string $s: E \rightarrow \{0, 1\}$, where $s(e) = 0$ for edge $e = (r, s) \in E$ represents the BP vacuum contribution from $|\mu_{r \rightarrow s}\rangle\langle\mu_{s \rightarrow r}|$ and $s_e = 1$ represents the “excited edge” contribution from the orthogonal projector $\mathcal{P}_{rs}^\perp$. Each configuration s defines an edge-induced subgraph [64] $G_s \subset G$ of excited edges. This results in,

$$\mathcal{Z} = Z_0 + \sum_{s \neq \mathbf{0}} Z_s, \quad (8)$$

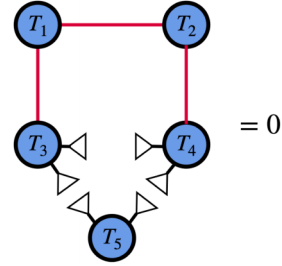
where Z_s denotes the contribution from configuration s normalized by the vacuum contribution. Now, we note that any configuration s which has an dangling edge will vanish. The “vacuum” contribution is Z_0 .

Definition II.1. (Generalized loops). Consider a graph $G = (V, E)$. A generalized loop is subgraph $C = (W, F)$ with $W \subseteq V$, $F \subseteq E$, with the property that the degree of any $w \in W$ in C is at least two. The weight of a generalized loop is defined as the number of edges $|F|$.

We denote the set of generalized loops in graph G as $\mathcal{L}_G$. Note that a generalized loop need not be a simple loop or even a connected subgraph. With mild abuse of notation, we refer to generalized loops simply as “loops” and specify “simple loops” when needed. We denote a loop as $l \in \mathcal{L}_G$ with loop weight $|l|$.

Lemma II.1. A non-zero excitation Z_s is possible only if G_s is a generalized loop in G [50,52].

Hence, contributions such as the following with a “dangling excitation” vanish,



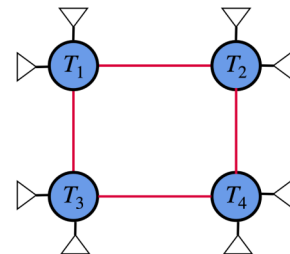
owing to T_3 and T_4 having degree one. Thus, it is possible to write the series expansion involving only those configurations which form generalized loops. Let us first define formally a *loop correction*.

Definition II.2. (Loop correction). Let l be a generalized loop in the graph. Each loop l has an associated loop correction $Z_l \in \mathbb{C}$, defined as

$$Z_l = \left(\prod_{(w,v) \in V(l)} \mathcal{P}_{wv}^\perp \right) \star \left(\prod_{(w,v) \notin V(l)} \mu_{wv} \otimes \mu_{vw} \right) \star \left(\prod_v T_v \right), \quad (9)$$

where $V(l)$ is the set of vertices in the loop l .

For instance, a simple loop correction on the loop $l = \{(1, 2), (2, 3), (3, 4), (1, 4)\}$ on a square lattice is,



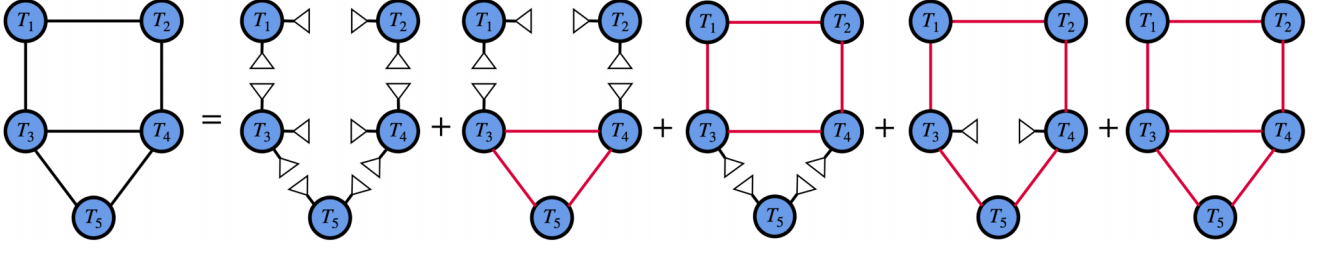


FIG. 1. Loop series expansion. The contraction of a five-vertex tensor network can be exactly represented as the sum of the BP vacuum and all the generalized loop excitations on the graph. The red line denotes the orthogonal projector as defined in Eq. (7).

where the message tensors from all other connected vertices are contracted in.

The loop series expansion is then given as follows.

Lemma II.2. (Loop series expansion). Consider the expansion of the partition function of the tensor network $T = (\{T_v\}_{v \in V}, V, E)$ by resolving the identity at each edge. Then, we have that

$$\mathcal{Z}(T) = Z_0 + \sum_{l \in \mathcal{L}_G} Z_l, \quad (10)$$

where the only non-zero excited contributions are generalized loops l in $G = (V, E)$.

Crucially, the sum in Lemma II.2 runs over all possible loops, including disconnected ones. We illustrate this expansion for a simple tensor network consisting of five vertices in Fig. 1. The net contraction is given as a sum of the BP vacuum along with all possible edge excitations, which appear as generalized loops.

D. Divergence due to disconnected loops

Ideally, one would like to approximate the loop expansion by truncating the series at some finite loop weight. Specifically, we set a cut-off weight and sum over loops with weights $\leq m$. Denoting this by $\hat{Z}_m$, we have,

$$\hat{Z}_m = Z_0 + \sum_{\substack{l \in \mathcal{L}_G \\ |l| \leq m}} Z_l \quad (11)$$

Empirically, one expects that if Z_l decays exponentially fast with l , then $\hat{Z}_m$ provides a good approximation to $\mathcal{Z}(T)$ with a modest cutoff m . A major problem with this

approach is the need to sum over generalized loops that are disconnected. To see this, consider a tensor network (PEPS) defined on a $L \times L$ two-dimensional square lattice. On regular lattices such as the 2D square lattice, there are combinatorially many disconnected loops. For example, there are order $\binom{L^2}{2}$ disconnected loops of weight 8 formed by two loops of weight 4 on plaquettes [Fig. 2(a)]. Going to higher weights, there are order $\binom{L^2}{3}$ disconnected loops formed by three connected loops, and so on. Therefore, as one goes to higher m , the number of terms grows combinatorially, outweighing the exponential decay of individual terms. This represents a significant bottleneck to the convergence of the loop series expansion, and motivates us to look for improved techniques.

III. CLUSTER EXPANSION

In this section, we introduce the cluster expansion and prove the main technical result: the tensor cluster expansion converges if loops Z_l decay exponentially in $|l|$ with a sufficiently large exponent. Crucially, the cluster expansion technique overcomes the challenge of disconnected loops. We first give a physical picture about why cluster expansion provides a better series expansion than the loop series expansion. We then introduce the cluster expansion formally and state the main result. We give a toy example and compare with earlier work in the end.

A. Physical intuition behind cluster expansion

As we have argued in the previous section, tensor network contractions can be thought of as generalizations of partition functions in statistical mechanics by adding

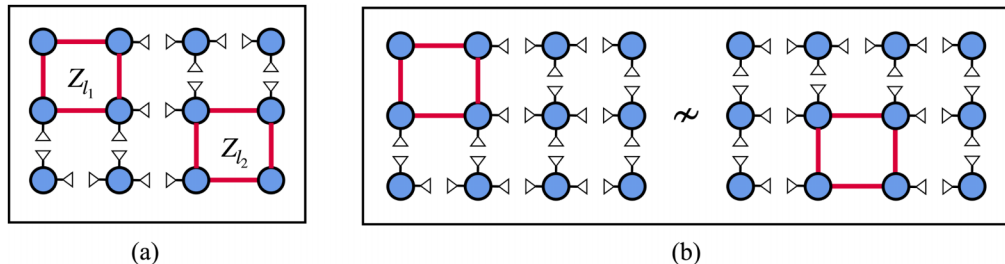


FIG. 2. (a) Disconnectedness and incompatibility. Example of a disconnected loop. (b) Example of two incompatible loops.

sign structures. Now, partition functions are not stable objects under local perturbations. For example, by heating any one site to infinite temperature, the partition function is changed by a constant *multiplicative* factor. To account for this in a series expansion, each site must show up in a *constant fraction* of terms. This is fundamentally why the loop series necessitates the use of the disconnected loops and the combinatorial growth of the number of terms in the loop expansion of $\mathcal{Z}$.

On the other hand, the free energy $\mathcal{F}$ is a well-behaved object under local perturbations. When one site is heated to infinite temperature, the free energy is changed only by a constant additive factor. This behavior is realizable in series expansions without disconnected objects: given a fixed site, only a *constant number* of terms should be involved in the expansion. Therefore, one naively expects that the series expansion of the free energy is better behaved.

Another related intuition regarding the cluster expansion is the presence of non-linearity in the series to ensure convergence. As we will see, the non-linearity in the loop contributions added through the cluster method leads to provable convergence.

B. Formal definition

Now we formally introduce the cluster expansion. In particular, we employ the formalism of the abstract polymer model [65] which provides black-box techniques to prove convergence. Throughout this subsection, we will work with the normalized tensor $\tilde{T}_v$ defined as follows,

$$\tilde{T}_v = \frac{T_v}{Z^{(v)}}, \quad (12)$$

where $Z^{(v)}$ is the local contribution defined in Eq. (4). Under this normalization, the BP contraction of $\tilde{T}_v$ is one, and correspondingly the BP free energy of $\tilde{T}_v$ is zero. We will compute the cluster expansion of $\mathcal{F}(\tilde{T})$, which is related to $\mathcal{F}(T)$ by a constant offset, namely,

$$\mathcal{F}(T) = \mathcal{F}(\tilde{T}) + \sum_v \log(Z^{(v)}). \quad (13)$$

Crucially, the loop series expansion of $\mathcal{Z}(\tilde{T})$ contains the contributions from all generalized loops, which includes connected as well as disconnected loops. We term the disconnected part as consisting of *compatible* loops, in the following sense. Denote $\mathcal{L}_G^c$ to be the set of connected loops in the graph G .

Definition III.1. (Compatible loops). Two loops l, l' in $\mathcal{L}_G^c$ are said to be compatible, written $l \sim l'$, if they do not overlap; that is, they share no vertex or edge in the underlying graph. A family $\Gamma \subset \mathcal{L}_G^c$ of loops is called

compatible if every pair of distinct loops in Γ is compatible.

We give a pair of incompatible loops in Fig. 2(b). We note that the notion of *loop compatibility* is conceptually similar, though not identical, to the notion of *connectedness* of loops. Connectedness is a property of a single loop, describing whether it can be decomposed into two disconnected subgraphs. In contrast, loop compatibility is a relation between two distinct loops and does not depend on their individual connectedness. If two loops are compatible, their union forms a larger but disconnected loop.

Next, we define the object *cluster*. Intuitively, clusters are a collection of loops, which can be described as a multiset as follows.

Definition III.2. (Clusters). A cluster is a collection of tuples of the form

$$\mathbf{W} = \{(l_1, \eta_1), (l_2, \eta_2), \dots, (l_m, \eta_m)\},$$

where each $l_i \in \mathcal{L}_G^c$ is a loop and η_i is the multiplicity of the loop l_i in the cluster. The total number of loops in the cluster is denoted as $n_{\mathbf{W}} = \sum_{i=1}^m \eta_i$.

We define the cluster weight $|\mathbf{W}| = \sum_i \eta_i |l_i|$, where $|l_i|$ is the weight of loop l_i . We also denote $\mathbf{W}! = \prod_i \eta_i!$. We denote the correction of the cluster $Z_{\mathbf{W}}$ as the product of the loop corrections raised to their respective multiplicities:

Definition III.3. (Cluster correction). For a cluster $\mathbf{W} = \{(l_1, \eta_1), (l_2, \eta_2), \dots, (l_m, \eta_m)\}$, the cluster correction is defined as

$$Z_{\mathbf{W}} = \prod_{i=1}^m Z_{l_i}^{\eta_i}, \quad (14)$$

where Z_{l_i} are the corresponding loop corrections.

Given a cluster $\mathbf{W}$, we define the *interaction graph* as follows.

Definition III.4. (Interaction graph). Given a cluster $\mathbf{W} = \{(l_1, \eta_1), (l_2, \eta_2), \dots, (l_m, \eta_m)\}$, we define the interaction graph $G_{\mathbf{W}} = (V_{\mathbf{W}}, E_{\mathbf{W}})$ as follows: each loop l_i corresponds to η_i distinct vertices, with the total number of vertices being $|V_{\mathbf{W}}| = \sum_{i=1}^m \eta_i$. There is an edge $(l, l') \in E_{\mathbf{W}}$ either if the loops l and l' are incompatible $l \not\sim l'$, or they are identical $l = l'$.

A cluster $\mathbf{W}$ is called *connected* if its interaction graph $G_{\mathbf{W}}$ is connected—that is, there exists a path between any two vertices in $G_{\mathbf{W}}$. This connectivity condition is crucial: we now establish that only connected clusters contribute to the free energy expansion.

Lemma III.1. (Connected clusters only). The free energy can be expressed as

$$\mathcal{F}(\tilde{T}) = \sum_{\text{connected } \mathbf{W}} \phi(\mathbf{W}) Z_{\mathbf{W}}, \quad (15)$$

where the sum runs over all connected clusters $\mathbf{W}$. The coefficient $\phi(\mathbf{W})$ is called the Ursell function, given as

$$\phi(\mathbf{W}) = \begin{cases} 1, & \eta_{\mathbf{W}} = 1 \\ \frac{1}{\mathbf{W}!} \sum_{\substack{C \in G_{\mathbf{W}} \\ C \text{ connected}}} (-1)^{|E(C)|}, & \eta_{\mathbf{W}} > 1 \end{cases} \quad (16)$$

where C is a connected subgraph of the interaction graph $G_{\mathbf{W}}$ spanning all vertices, and $E(C)$ is the edge set of C .

The proof of Lemma III.1 is rather technical so we defer it to the appendix. Since the expansion involves only connected clusters, an important question arises: how many connected clusters must be enumerated in the truncated sum? The answer depends fundamentally on the graph structure, as quantified by the following combinatorial bound.

Lemma III.2. Given any graph with n vertices and with degree Δ , the number of connected clusters with weight $\leq m$ is upper-bounded by $n(e(\Delta + 2))^{2m}$.

We prove this bound in the SM [66]. This bound reveals that there are $n\Delta^{O(m)}$ connected clusters per site, making the enumeration computationally tractable for moderate values of m . We also note that the base coefficient is not optimal since we only use elementary techniques to give a crude bound. Given this bound, we are then motivated to *truncate* the cluster series at a finite cluster weight m to get the truncated free energy $\tilde{F}_m$.

Definition III.5. (Truncated cluster expansion). The truncated partition function $\tilde{F}_m$ is defined as

$$\tilde{F}_m = \sum_{\substack{\text{connected } \mathbf{W} \\ |\mathbf{W}| \leq m}} \phi(\mathbf{W}) Z_{\mathbf{W}}. \quad (17)$$

We will use the $\tilde{F}_m$ to approximate $\mathcal{F}(\tilde{T})$. Our main technical result is to show that when the loop contribution decays exponentially with their weight with an exponent above a constant threshold, then the cluster expansion converges and $\tilde{F}_m$ gives a good approximation to $\mathcal{F}(\tilde{T})$.

Theorem III.1. (Convergence of the cluster expansion). (Informal) Given a tensor network with degree Δ and normalized by the BP fixed point. Assume there exists a constant $c > c_0 = \Theta(\log(\Delta))$ such that

$$|Z_l| \leq e^{-c|l|}. \quad (18)$$

Then the series for $\mathcal{F}$ converges absolutely, and the error in truncating the series at order m is bounded by

$$|\mathcal{F} - \tilde{F}_m| = O(ne^{-d(m+1)}), \quad (19)$$

where $d = c - c_0$.

We prove our main result in the SM [66], which is a direct application of the Kotecký-Preiss condition [65]. If we denote the true free energy density as $f = \mathcal{F}/n$ and the weight- m cluster approximation density as $\tilde{f}_m = \tilde{F}_m/n$, then we have that sufficient loop decay ensures,

$$|f - \tilde{f}_m| = O(e^{-d(m+1)}). \quad (20)$$

We discuss the implications of our main theorem, summarized as the following corollary.

Corollary III.1. (Informal) Under the conditions of Theorem 1,

1. BP error in free-energy density is bounded by $|f - f_{\text{BP}}| = O(e^{-(c-c_0)})$, and,
2. Computing $\tilde{F}_m$ with $m = O(\log(n))$ gives a polynomial-time algorithm to approximate $\mathcal{F}$ up to inverse polynomial error.

At $m = 0$, our theorem tells that BP approximates the free energy density up to a constant additive error, under the stated assumption. Further, cluster expansion improves this error exponentially fast in m . Hence, to get to an error ϵ , one needs to enumerate clusters to order $m = (1/d) \log(1/\epsilon)$. Moreover, all connected clusters with weight $\leq m$ can be enumerated in $n e^{O(m)} = O(n/\epsilon^{1/d})$ time. Thus, the time complexity is $\text{poly}(n)$ to get to an inverse polynomial error in free energy density.

Finally, in certain cases such as simulating quantum dynamics, the tensor network contraction $\mathcal{Z}(T)$ itself is the physical observable, and we are interested in quantifying its error. An *additive* error of ϵ in $\mathcal{F}(T)$ corresponds to a *multiplicative* error of $\Theta(\epsilon)$ in $\mathcal{Z}(T)$. Since these observables are often of order $O(1)$, this typically implies an $O(1)$ additive error. However, when the observables become exponentially small, an $O(1)$ additive error is no longer meaningful. In contrast, a $\Theta(\epsilon)$ multiplicative error ensures that the additive error bar shrinks proportionally as the observable decreases. This makes the cluster expansion particularly favorable in such regimes.

C. Toy example

To illustrate the idea behind the cluster expansion and compare it to the loop expansion, we consider a toy example. Consider a tensor network on a one-dimensional ring. The only generalized loop is the entire ring l , and the loop contribution is Z_l . Suppose we have normalized the tensor network by the BP fixed point, so the BP contribution is one. Then, the loop expansion gives

$$\mathcal{Z} = 1 + Z_l \quad (21)$$

and the free energy is

$$\mathcal{F} = \log(1 + Z_l). \quad (22)$$

On the other hand, the only possible clusters are $\{(l, 1)\}, \{(l, 2)\}, \dots$, namely the same loop repeated multiple times. In this case, all clusters of weight m are $\{(l, m)\}$. Therefore, $\mathbf{W}! = m!$. The part that sums over connected spanning graphs evaluates to $(-1)^{m+1}(m-1)!$ which we illustrate in the SM [66]. Therefore, the Ursell function can be computed to be $\phi(\{(l, m)\}) = \frac{(-1)^{m+1}}{m}$. The cluster expansion gives

$$\mathcal{F} = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} Z_l^k. \quad (23)$$

This is exactly the Taylor expansion of $\log(1 + Z_l)$, which converges for $|Z_l| < 1$.

We note that in this small example, loop expansion converges in the first order, while the cluster expansion needs to go to higher orders. However, when $|Z_l|$ is small, both methods agree on the leading order. As we will see later, when contracting large tensor networks, we expect cluster expansion to converge faster. We also note that while the computation of the Ursell function is daunting here, it is drastically simplified in large tensor networks since one typically does not need to handle clusters with many loops. In fact, in the numerical work in the next section, the Ursell function can be brute-force enumerated in the order we truncate.

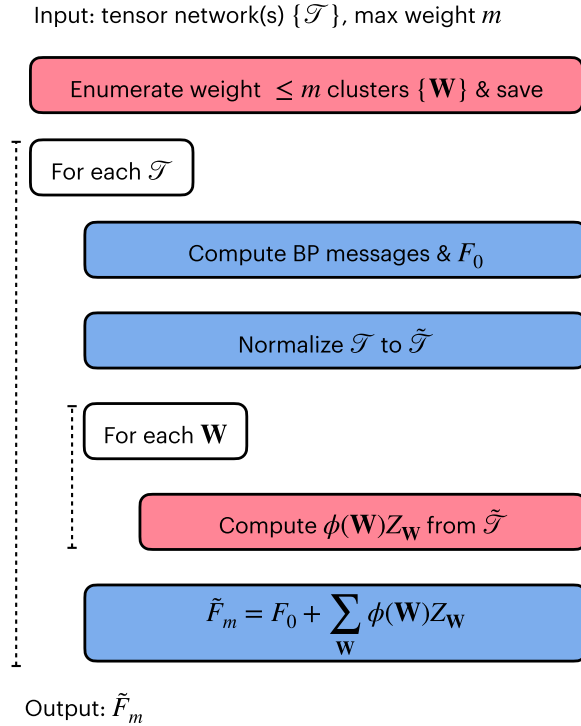


FIG. 3. Pseudocode of cluster expansion. Computationally expensive steps are colored in red. When there are more than one tensor networks, we assume they are defined on the same graph so that they share the same clusters.

Finally, this toy example also illustrates the difference between the linked cluster expansion and our cluster expansion. The linked cluster expansion produces $\log(1 + Z_l)$ in one iteration [61], whereas our cluster expansion is an infinite series expansion.

IV. ALGORITHMS

In this section, we present an overview of the algorithmic procedure for computing the cluster expansion of generic tensor networks. Figure 3 provides a pseudocode summary. Suppose we are given a family of tensor networks $\{\mathcal{T}\}$ defined on a common graph G , and our goal is to contract each network. The algorithm takes as input the set $\{\mathcal{T}\}$ and a maximal cluster weight m , and outputs the truncated cluster expansion $\tilde{F}_m$ for each $\mathcal{T}$.

A. Cluster enumeration

The first step is to enumerate all connected clusters with weight $\leq m$ and save them. This step is computationally expensive, as its complexity grows exponentially with m . However, for a given graph G , this computation only needs to be performed once. The cluster enumeration algorithm is intricate; therefore, we provide a detailed discussion in the SM [66]. Here, we summarize the main steps:

1. For each vertex, enumerate all connected loops with weight $\leq m$ supported on that vertex.
2. Repeat over all vertices to obtain a list of connected loops, then deduplicate to remove redundancies.
3. For each vertex, enumerate all connected clusters with weight $\leq m$ supported on that vertex, using the list of connected loops.
4. Repeat over all vertices to obtain a list of connected clusters, then deduplicate.

For step 1, we use a depth- or breadth-first search algorithm to “grow” a connected subgraph from each vertex, recording the subgraph only when it forms a generalized loop (see Definition II.1). Repeating this process over all vertices yields a list of connected loops. Since a single loop may be supported on multiple vertices, we perform de-duplication to remove redundancies. The deduplication in step 2 can be efficiently implemented by storing a hash map of the loop objects. Every time a new loop is found, one simply query the hash map and add the loop only if the loop is not in the hash map.

Step 3 follows a similar approach: starting from each vertex, we “grow” connected clusters using the list of loops, and again deduplicate after repeating over all vertices. Finally, we repeat step 3 for each vertex and deduplicate to obtain the final list of connected clusters. As in step 2, this deduplication can be performed on the fly by maintaining a hash map.

Steps 1 and 3 have a runtime of $O(\exp(m))$, while steps 2 and 4 add an additional factor of n [70]. Thus, the total runtime is $O(n \exp(m))$. In practice, steps 1 and 2 are the

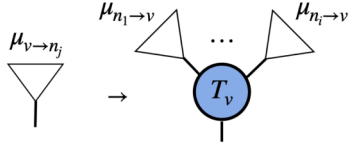
most time-consuming. The supplementary material [66] discusses strategies to improve runtime, such as exploiting symmetries (such as translational symmetry) of the graph G .

B. Message passing and normalization

After enumerating all connected clusters, we proceed to compute the cluster expansion for each tensor network $\mathcal{T}$. The next step is to run BP on $\mathcal{T}$ to obtain the (approximate) fixed-point messages $\{\mu_{v \rightarrow s}\}$. This is achieved by iteratively applying the following update rule until convergence:

$$\mu_{v \rightarrow s} \rightarrow \left(\bigotimes_{r \in \mathcal{N}(v)/\{s\}} \mu_{r \rightarrow v} \right) \star T_v. \quad (24)$$

Schematically,



For numerical stability, we normalize the message vector by its two-norm after each update. This does not affect the result in the end as any constant factor will be canceled out when dividing by I_{vw} . Message passing is typically efficient, as belief propagation often converges rapidly. However, tensor networks with sign structures may either fail to converge or converge slowly. Additionally, message passing can admit multiple fixed points, some of which may yield poor approximations.

Empirically, we find that the following strategies are helpful: (1) initializing messages randomly (in case of symmetry breaking), (2) *damping* the message passing—i.e., updating messages as a convex combination of the old and new values, and (3) adding a small amount of noise to the messages at each iteration. It is imperative to ensure that the self-consistency condition is satisfied to some fixed error ε at each vertex, that is, $\forall v \in V, s \in \mathcal{N}(v)$ we ensure,

$$\left\| \left(\bigotimes_{r \in \mathcal{N}(v)/\{s\}} \mu_{r \rightarrow v} \right) \star T_v - \mu_{v \rightarrow s} \right\|_2 < \varepsilon. \quad (25)$$

Once the messages have converged, we compute the BP free energy F_0 and normalize the tensors $\mathcal{T}$ to $\tilde{\mathcal{T}}$ as in Eq. (12). This normalization introduces a constant offset in the free energy, so we add F_0 back to the final result.

C. Computing cluster contribution and final result

The final step is to load the list of connected clusters and compute their contributions $Z_{\mathbf{W}}$ as described in Eq. (14).

This step is another computational bottleneck, since the number of clusters grows exponentially with m . However, it is highly parallelizable, as the contribution of each cluster can be computed independently. For graphs with significant symmetries (e.g., square lattices), many loops share the same shape, allowing them to be batched together for efficient graphical processing unit acceleration.

Each cluster requires contracting relatively small loop tensors. When the bond dimension is small, this is efficient. However, when the bond dimension is large (which could happen in the context of simulating quantum dynamics), optimizing the contraction order is necessary. The Ursell function $\phi(\mathbf{W})$, defined in Eq. (16), can in principle be precomputed, but in practice it is often straightforward to evaluate. For example, in PEPS, the Ursell function is always 1, -1 , or $-1/2$ for weights up to twelve. Finally, we sum over all connected clusters with weight $\leq m$ to obtain the truncated cluster expansion. The final result is given by $F_0 + \sum_{\mathbf{W}} \phi(\mathbf{W}) Z_{\mathbf{W}}$.

V. BENCHMARK: 2D ISING MODEL

To demonstrate the efficacy of tensor network belief propagation and validate the cluster expansion methodology, we apply the framework to the paradigmatic two-dimensional classical Ising model. This system serves as an ideal testbed for several reasons: it possesses a known exact solution due to Onsager [71], exhibits a second-order phase transition with well-characterized critical behavior, and the partition function can be naturally formulated as a tensor network with constant bond dimension. The numerical calculations which follow have been performed using the ITensor [72] library.

We consider the classical 2D Ising model on a $L \times L$ square lattice with $N = L^2$ spins and nearest-neighbor interactions, as described by the Hamiltonian,

$$H[\{s_i\}] = -J \sum_{\langle ij \rangle} s_i s_j, \quad (26)$$

where $s_i \in \{\pm 1\}$ denotes the spin variable at site i , $J > 0$ is the ferromagnetic coupling strength, and the sum runs over all nearest-neighbor pairs $\langle ij \rangle$ on a square lattice with periodic boundary conditions. The partition function is given by

$$Z = \sum_{\{s_i\}} \exp \left(\beta \sum_{\langle ij \rangle} s_i s_j \right),$$

where $\beta = 1/(k_B T)$ is the inverse temperature.

To represent this partition function as a tensor network, we use a factorization of the Boltzmann weights. For each nearest-neighbor interaction, we use the identity

$$e^{\beta s_i s_j} = \sum_{x=0}^1 w(s_i, x, \beta) w(s_j, x, \beta),$$

where the function $w(s, x, \beta)$ is defined as

$$w(s, x, \beta) = \begin{cases} \sqrt{\cosh(\beta)}, & x = 0, \\ \sqrt{\cosh(\beta)} \cdot s \cdot \sqrt{\tanh(\beta)}, & x = 1. \end{cases}$$

We then define a rank-4 local tensor $T_{x_1 x_2 x_3 x_4}$ at each lattice site, corresponding to the four directions (up, down, left, right), as follows:

$$T_{x_1 x_2 x_3 x_4} = \sum_{s=\pm 1} w(s, x_1, \beta) \cdot w(s, x_2, \beta) \cdot w(s, x_3, \beta) \cdot w(s, x_4, \beta).$$

The full partition function is then given by contracting these local tensors according to the 2D square lattice geometry:

$$Z = \sum_{\{x_{i,j}\}} \prod_{\text{sites } (i,j)} T_{x_{i,j}^{(u)} x_{i,j}^{(d)} x_{i,j}^{(l)} x_{i,j}^{(r)}},$$

where each index $x_{i,j}^{(\cdot)}$ is shared with the corresponding neighboring site, and the sum is over all internal bond indices $x_{i,j}^{(\cdot)} \in \{0, 1\}$, resulting in a bond dimension $\chi = 2$. The tensor network contraction of this ensemble produces the full partition function Z , from which thermodynamic quantities such as the free energy density $f = -\beta^{-1} \ln(Z)/N$ can be extracted.

A. BP vacuum

An important question underlying any approximation scheme is understanding the regimes where it provides reliable results. For BP on tensor networks, this translates

to identifying the physical conditions under which the BP vacuum accurately captures the system's behavior. Since BP implements a tree-like mean-field treatment—approximating each site's environment with a rank-one tensor—we expect it to perform well when mean-field assumptions are valid: deep within a phase and away from critical points.

Figure 4(a) tests this expectation by comparing the BP vacuum solution (dotted blue line) with Onsager's exact result (solid black line) for the free energy density $f(\beta)$ in the region $\beta \in [0.25, 0.55]$. The BP approximation indeed demonstrates good accuracy in both the high-temperature paramagnetic phase ($\beta \ll \beta_c$) and the low-temperature ferromagnetic phase ($\beta \gg \beta_c$), where deviations from the exact solution remain modest. However, significant discrepancies emerge near the critical point encompassing the phase transition, precisely where mean-field theory is expected to break down due to a diverging correlation length and enhanced fluctuations.

This behavior can be further understood through the theoretical foundations of the BP approximation. The BP vacuum solution implements the Bethe approximation, treating the square lattice as a locally tree-like structure by neglecting loop correlations. This is owing to the fact that BP fixed points are stationary points of the Bethe free-energy [73]. This mean-field-like treatment captures the essential physics away from criticality, where local correlations dominate, but becomes increasingly inaccurate near the phase transition where long-range fluctuations and thereby loop effects become significant.

For the Ising model on a Bethe lattice with coordination number z , the critical point occurs at $\beta_c^{(z)} = 0.5 \ln \frac{z}{z-2}$ [74]. Since the square lattice has $z = 4$, the BP critical point is located at $\beta_{\text{BP}} = \frac{\ln(2)}{2} \approx 0.347$, which we can verify through the divergence of the specific heat computed from the BP vacuum free energy. This BP critical point lies below the

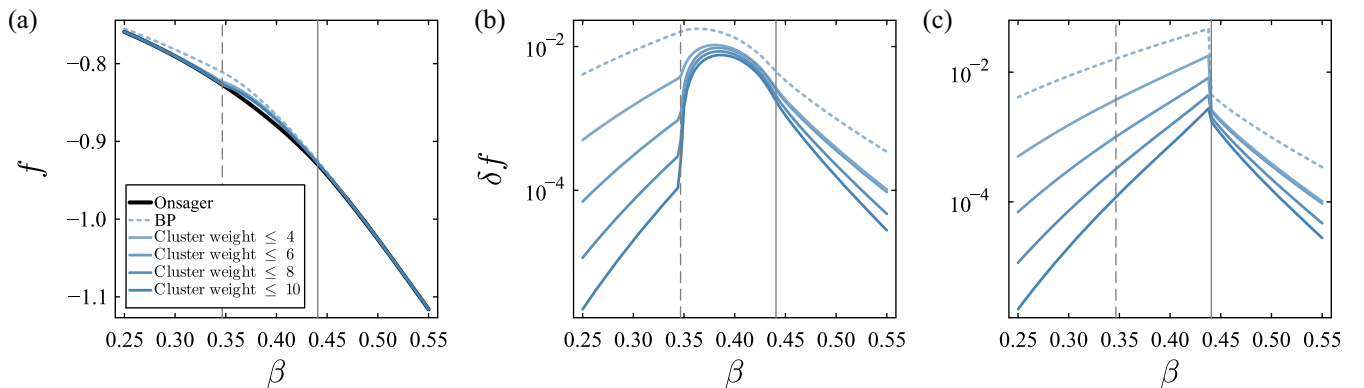


FIG. 4. Ising free energy. (a) Comparison of BP vacuum solution and subsequent cluster corrections evaluated using the message-passing fixed-point to free-energy density $f(\beta)$ in the region $\beta \in [0.25, 0.55]$ against the Onsager solution. The BP critical point is identified at $\beta_{\text{BP}} = \ln 2/2$ (dashed gray line) and 2D Ising critical point $\beta_c = \ln(1 + \sqrt{2})/2$ (solid gray line). (b) Free energy density error $\delta f(\beta)$ for BP vacuum (dashed) and different cluster corrections of weight $w \in \{4, 6, 8, 10\}$ using message-passing fixed-point. (c) Free energy density error of BP and cluster expansions evaluated with the symmetry-respecting fixed point.

true Onsager critical point $\beta_c = \ln(1 + \sqrt{2})/2 \approx 0.441$, explaining why BP accuracy deteriorates well before the actual phase transition.

B. Cluster expansion

To systematically improve upon the BP approximation, we implement the cluster expansion formalism by incorporating cluster corrections of increasing weight to the BP vacuum. Figure 4(a) presents a detailed view of free energy density in the region $\beta \in [0.25, 0.55]$, showing the progressive convergence toward the exact solution as cluster corrections of weight $w \in \{4, 6, 8, 10\}$ are added to the BP vacuum. The two critical temperatures of interest, β_{BP} and β_c are shown as dashed-gray and solid-gray lines in the plot. A clear discrepancy between BP and the exact solution appears in the intermediate regime $\beta_{BP} < \beta < \beta_c$.

The cluster corrections exhibit distinct convergence behavior across different temperature regimes. Deep in both the high-temperature and low-temperature phases, the cluster-corrected free energy rapidly converges to the exact result, with higher-weight contributions becoming quickly negligible. The convergence properties are further illuminated in Fig. 4(b), which displays the free energy density error $\delta f(\beta) = f_{\text{approx}}(\beta) - f_{\text{exact}}(\beta)$ for the BP vacuum (dashed) and successive cluster corrections of weight $w \in \{4, 6, 8, 10\}$. In both regimes $\beta < \beta_{BP}$ and $\beta > \beta_c$, the error decreases rapidly with increasing cluster weight, indicating efficient convergence of the expansion.

In contrast, the intermediate regime $\beta_{BP} < \beta < \beta_c$ exhibits markedly slower convergence. Here, BP message-passing selects a symmetry-broken fixed point on the Bethe lattice, while the true 2D system remains in the paramagnetic phase. This mismatch in fixed-point symmetry leads to poor expansion behavior when expanding around the message-passing solution, a phenomenon we refer to as the “confusion regime.” Expanding around the appropriate (symmetric) fixed point in this regime restores uniform convergence across the high-temperature phase, with the only breakdown occurring at the critical point itself. This is shown in Fig. 4(c), thus highlighting the crucial dependence of the cluster expansion algorithms on the choice of the fixed-point. Hence, the symmetry-respecting fixed-point is identical to the one obtained with message-passing for $\beta < \beta_{BP}$ and $\beta > \beta_c$, however must be the paramagnetic fixed-point in the confusion regime $\beta \in [\beta_{BP}, \beta_c]$. The emergence of multiple fixed-points is further discussed in the SM [66].

C. Convergence: Clusters vs loops

We now present the central numerical evidence of this work: a systematic comparison between our cluster expansion method and the traditional loop series expansion, revealing fundamental differences in how the

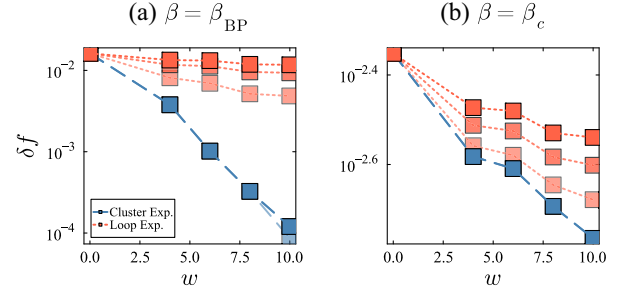


FIG. 5. Loop and cluster expansions. Free-energy density error δf at (a) $\beta = \beta_{BP}$ and (b) $\beta = \beta_c$ for cluster corrections (dashed, blue) and loop corrections (dotted, red) as a function of cluster (loop) weight w for system sizes $L \in \{10, 20, 30\}$ (decreasing transparency). Curves of cluster expansion at different L collapse because cluster expansion is automatically in the thermodynamic limit.

algorithms scale. Figure 5 compares the approximation errors of cluster corrections (dashed, blue) and loop corrections (dotted, red) as functions of correction weight $w \in \{4, 6, 8, 10\}$ for multiple system sizes $L \in \{10, 20, 30\}$. The results are plotted for (a) $\beta = \beta_{BP}$ and (b) $\beta = \beta_c$, expanded around the symmetry-respecting fixed-point.

The results reveal that while the cluster expansion exhibits robust exponential convergence that remains stable across all system sizes, the loop series expansion suffers from fundamental instabilities: (i) significantly slower convergence for any fixed system size, and (ii) divergent behavior as the system size increases, rendering it unsuitable for thermodynamic calculations.

This pathological behavior of loop expansions has a clear mathematical origin. Consider the BP vacuum contribution $Z_0 = z_0^N$ and a normalized weight- w loop contribution $Z_w = O(1)$ that is intensive in the system size (normalized). The loop series expansion computes the free energy density as

$$\frac{1}{N} \log[Z_0(1 + NZ_w)] = \log z_0 + \frac{1}{N} \log(1 + NZ_w). \quad (27)$$

In the thermodynamic limit $N \rightarrow \infty$, the correction term $(1/N) \log(1 + NZ_w) \rightarrow 0$, causing the loop contributions to vanish and negating any systematic improvement.

In contrast, the cluster expansion computes the free energy density as

$$\frac{1}{N} [\log Z_0 + NZ_w] = \log z_0 + Z_w. \quad (28)$$

This is automatically in the thermodynamic limit in the sense that the estimated free energy density does not depend on the system size. This fundamental difference ensures that cluster corrections provide stable, size-independent improvements to the BP approximation,

establishing the theoretical superiority of our tensor network-based approach for systematic corrections to belief propagation.

D. Loop contribution analysis and convergence properties

An important question in the application of BP is the identification of parameter (temperature) regimes where such corrections on top of BP vacuum become most significant, thereby delineating the domains of validity for the BP vacuum approximation. To further address this question, we analyze the average normalized loop contributions $Z_w(\beta)$ as functions of inverse temperature for loops of varying weight w .

Figure 6(a) presents the temperature dependence of average loop contributions Z_w for $w \in \{4, 6, 8, 10\}$ evaluated using the message-passing (solid lines) and the high-temperature fixed-point (dotted lines). For loops evaluated using the message-passing fixed-point, the contributions exhibit a maxima precisely at the BP critical point $\beta_{BP} = \ln 2/2$. This behavior provides compelling evidence that loop effects are intimately related to the correlations in the system, as the peak is at critical point of the Ising model on the $z = 4$ Bethe lattice β_{BP} .

When loops are evaluated around the paramagnetic fixed point, their magnitude grows monotonically as the temperature is lowered. In the intermediate “confusion regime,” however, care is required in interpreting this growth. Although loops constructed from the message-passing fixed point may exhibit smaller average contributions than in the high-temperature phase, the symmetry-breaking in the message-passing fixed-point introduces a systematic bias: the “all-edges-excited” loop sector acquires a finite $O(1)$ weight. This reflects a mismatch between the symmetry of the expansion point and that of the true Gibbs state. As a result, loop decay is

violated, and the cluster expansion fails to converge [as seen in Fig. 4(c)]. In contrast, expanding around the paramagnetic fixed point, which preserves the $\mathbb{Z}_2$ symmetry in this regime, restores decay of loop contributions and yields a well-behaved expansion [as seen in Fig. 4(d)].

Deep in either phase, loop contributions remain relatively modest, indicating that the BP vacuum provides a robust zeroth-order approximation in thermodynamically stable phases. However, approaching the critical point, the amplification of loop effects signals the need for systematic inclusion of higher-order corrections to achieve quantitative accuracy. As we have demonstrated, this requirement is optimally addressed by the cluster expansion method, which exhibits exponential convergence and thermodynamic stability in precisely this challenging regime.

Equally crucial for validating our theoretical framework is examining the exponential decay rate of loop contributions Z_w with increasing weight w . Figure 6(b) investigates this by analyzing the decay of loop contributions $Z_w(\beta)$ as a function of loop weight w at three representative temperatures: the high-temperature phase ($\beta = 0.25$), the low-temperature phase ($\beta = 0.55$), and the BP critical point ($\beta_{BP} = \ln 2/2$). These are evaluated for the symmetry-respecting fixed-point.

We observe rapidly decaying loop contributions at $\beta = 0.25$, reminiscent of the traditional high-temperature expansions. The decay curve at the BP critical point $\beta = \beta_{BP}$ shows larger loop magnitudes and slower decay, owing to the criticality of the model on the Bethe lattice. Finally, at low temperatures $\beta = 0.55$, the loop magnitudes are much smaller, yet they decay with a slower rate. This is owing to the symmetry-breaking in the messages, with the other symmetry sector occupying a constant weight. When evaluating symmetry-broken quantities (e.g., longitudinal magnetization) this slowdown is no longer a problem as the cluster-corrections operate effectively in one of the two ferromagnetic “valleys.”

These results indicate qualitatively that exponential convergence of the cluster expansion tracks the rapid decay of loop excitations. In contrast to the analytical bounds, we note that Theorem 1 operates in the worst-case regime: it assumes uniform decay $\sim e^{-c|l|}$ of *all* loops, while in practice loop tensors display additional structure (e.g., suppressed odd loops), leading to substantially improved behavior beyond the worst-case guarantee.

VI. DISCUSSION

We have presented a systematic theory of belief propagation for approximate tensor network contractions. Our main contribution is the construction of a cluster expansion that systematically improves upon the BP approximation. We rigorously prove that the cluster expansion converges exponentially fast if the loop contributions decay exponentially with a sufficiently large exponent. This result

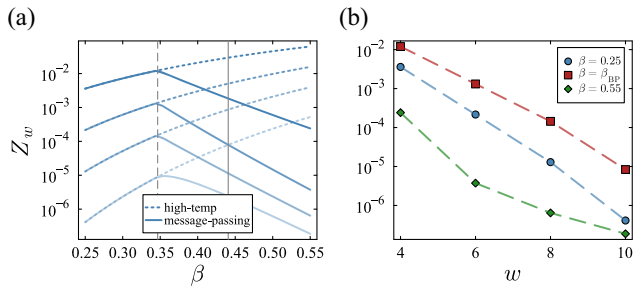


FIG. 6. Ising loop decay. (a) Average loop contributions $Z_w(\beta)$ for $w \in \{4, 6, 8, 10\}$ evaluated using the message-passing fixed-point (solid lines) and the high-temperature fixed-point (dotted) with the BP critical point identified at $\beta_{BP} = \ln 2/2$ (dashed gray line) and 2D Ising critical point $\beta_c = \ln(1 + \sqrt{2})/2$ (solid gray line). (b) Decay of average loop contributions of the symmetry-respecting fixed-point as a function of loop weight w in the low-temperature phase $\beta = 0.55$, high-temperature $\beta = 0.25$ and the BP critical point $\beta_{BP} = \ln 2/2$.

resolves two main challenges of BP: (1) it explains when BP provides a good approximation to the ground truth and (2) supplies an error estimate controlled by loop contributions. As a by product, it also yields a polynomial-time algorithm that systematically improves the BP result.

To bring this technique into practice, we present a detailed and optimized algorithmic procedure for computing the cluster expansion. We benchmark our algorithm against both BP and the loop series expansion in the two-dimensional Ising model. Our results show that while BP deviates from the exact solution near the critical point, the cluster expansion yields significant improvements. Moreover, we demonstrate, both numerically and analytically, that the loop expansion fails to correct BP in the thermodynamic limit. Notably, this happens even at leading order before the onset of combinatorial growth.

While the cluster expansion does not converge in adversarial cases (otherwise there will be severe complexity-theoretic consequences), we believe that it should converge in many physically-relevant setups. In particular, the convergence of the cluster expansion reflects that the system is short-range correlated. Therefore, incorporating small clusters would already account for most of the correlations. This idea is not new in the literature; many earlier works have exploited short-ranged correlations to simplify tensor network contractions, for example see [75,76]. An important theoretical question is to understand what kind of tensor networks admit a good BP approximation with a convergent cluster expansion.

Our work opens several avenues for future applications. First, BP is widely used in decoding classical and quantum error-correcting codes [31,77,78]. It would be interesting to explore the application of our method to improve the performance of BP-based decoders and provide rigorous error estimates. In particular, it is known that naive BP fails to give a threshold in quantum-LDPC codes because of the degeneracy problem [43]. Since the cluster expansion captures the short-range correlations omitted by the BP approximation, it may help restore a threshold, if not simply improving the decoding performance, which we investigate in a forthcoming work [79].

Second, many classical simulations of quantum systems and quantum dynamics heavily involve tensor networks, and more recently BP and its combination with other techniques [38] in studying quantum dynamics. It would be interesting to explore the application of our method to improve the accuracy of these simulations. In particular, given well established competing methods such as TEBD, it is important to understand the regimes where BP is advantageous and can be incorporated into the simulators toolbox.

Finally, it would be interesting to extend our method to other tensor network geometries, such as higher-dimensional regular lattices or random expander graphs, where many conventional approaches break down. Belief propagation is particularly well-suited to these settings,

as it is one of the few techniques that can handle arbitrary geometries while remaining computationally efficient. Moreover, BP is known to perform exceptionally well on locally tree-like structures, making it a promising tool for studying complex tensor networks beyond standard low-dimensional lattices.

Note added. After the completion of this work, a related work employing cluster expansion on tensor networks was posted [62]. See Sec. I A for detailed discussions.

ACKNOWLEDGMENTS

We thank Hongkun Chen, Rhine Samajdar, David Huse, Sarang Gopalakrishnan, Dmitry Abanin, Joseph Tindall, Will Staples, and Dries Sels for helpful discussions and collaborations on related projects. We are particularly indebted to Grace Sommers for extensive discussions and cross-checking some of our numerical results. We thank the anonymous reviewers for their feedback, helping improve the manuscript. This work was supported by a Brown Investigator Award, a program of the Brown Institute for Basic Sciences at the California Institute of Technology. Y.F.Z. acknowledges support from NSF QuSEC-TAQS OSI 2326767. The numerical simulations presented in this article were performed on computational resources managed and supported by Princeton University's Research Computing. Parhey's company is also acknowledged.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available. The data are available from the authors upon reasonable request.

-
- [1] Steven R. White, *Density matrix formulation for quantum renormalization groups*, *Phys. Rev. Lett.* **69**, 2863 (1992).
 - [2] Steven R. White, *Density-matrix algorithms for quantum renormalization groups*, *Phys. Rev. B* **48**, 10345 (1993).
 - [3] Ulrich Schöllwöck, *The density-matrix renormalization group in the age of matrix product states*, *Ann. Phys. (N.Y.)* **326**, 96 (2011).
 - [4] Román Orús, *A practical introduction to tensor networks: Matrix product states and projected entangled pair states*, *Ann. Phys. (N.Y.)* **349**, 117 (2014).
 - [5] J. Ignacio Cirac, David Perez-Garcia, Norbert Schuch, and Frank Verstraete, *Matrix product states and projected entangled pair states: Concepts, symmetries, theorems*, *Rev. Mod. Phys.* **93**, 045003 (2021).
 - [6] Frank Verstraete and J. Ignacio Cirac, *Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems*, *Adv. Phys.* **57**, 143 (2008).
 - [7] Guifré Vidal, *Entanglement renormalization*, *Phys. Rev. Lett.* **99**, 220405 (2007).

- [8] Norbert Schuch and Ignacio Cirac, *Computational complexity of projected entangled pair states*, *Phys. Rev. A* **76**, 012310 (2007).
- [9] Norbert Schuch, Michael M. Wolf, Frank Verstraete, and J. Ignacio Cirac, *Computational difficulty of finding matrix product ground states*, *Phys. Rev. Lett.* **98**, 140506 (2007).
- [10] Zeph Landau, Umesh Vazirani, and Thomas Vidick, *A polynomial time algorithm for the ground state of one-dimensional gapped local Hamiltonians*, *Nat. Phys.* **11**, 566 (2015).
- [11] Igor L. Markov and Yaoyun Shi, *Simulating quantum computation by contracting tensor networks*, *SIAM J. Comput.* **38**, 963 (2008).
- [12] Itai Arad, Alexei Kitaev, Zeph Landau, and Umesh Vazirani, *An area law and sub-exponential algorithm for 1D systems*, [arXiv:1301.1162](https://arxiv.org/abs/1301.1162).
- [13] Jonas Haferkamp, Dominik Hangleiter, Jens Eisert, and Marek Gluza, *Contracting projected entangled pair states is average-case hard*, *Phys. Rev. Res.* **2**, 013010 (2020).
- [14] Dylan Harley, Freek Witteveen, and Daniel Malz, *Computational complexity of injective projected entangled pair states*, [arXiv:2509.19963](https://arxiv.org/abs/2509.19963).
- [15] Guifré Vidal, *Efficient classical simulation of slightly entangled quantum computations*, *Phys. Rev. Lett.* **91**, 147902 (2003).
- [16] Guifré Vidal, *Efficient simulation of one-dimensional quantum many-body systems*, *Phys. Rev. Lett.* **93**, 040502 (2004).
- [17] M. Fannes, B. Nachtergaele, and R.F. Werner, *Finitely correlated states on quantum spin chains*, *Commun. Math. Phys.* **144**, 443 (1992).
- [18] D. Pérez-García, F. Verstraete, M. M. Wolf, and J. I. Cirac, *Matrix product state representations*, *Quantum Inf. Comput.* **7**, 401 (2007).
- [19] 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).
- [20] Judea Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Morgan Kaufmann Publishers Inc., San Francisco, CA, 1988).
- [21] Hans A. Bethe, *Statistical theory of superlattices*, *Proc. R. Soc. A* **150**, 552 (1935).
- [22] Alec Kirkley, George T. Cantwell, and M. E. J. Newman, *Belief propagation for networks with loops*, *Sci. Adv.* **7**, eabf1211 (2021).
- [23] C. Laumann, A. Scardicchio, and S.L. Sondhi, *Cavity method for quantum spin glasses on the Bethe lattice*, *Phys. Rev. B* **78**, 134424 (2008).
- [24] Michael Chertkov, *Exactness of belief propagation for some graphical models with loops*, *J. Stat. Mech.* (2008) P10016.
- [25] Matthew S. Leifer and David Poulin, *Quantum graphical models and belief propagation*, *Ann. Phys. (N.Y.)* **323**, 1899 (2008).
- [26] Elina Robeva and Anna Seigal, *Duality of graphical models and tensor networks*, *Inf. Inference* **8**, 273 (2019).
- [27] Robert G. Gallager, *Low-density parity-check codes*, *IRE Trans. Inf. Theory* **8**, 21 (1962).
- [28] David J. C. MacKay and Radford M. Neal, *Good error-correcting codes based on very sparse matrices*, *IEEE Trans. Inf. Theory* **45**, 399 (1999).
- [29] Hanwen Yao, Waleed Abu Laban, Christian Häger, Alexandre Graell i Amat, and Henry D. Pfister, *Belief propagation decoding of quantum LDPC codes with guided decimation*, in *Proceedings of the 2024 IEEE International Symposium on Information Theory (ISIT)* (IEEE, New York, 2024), pp. 2478–2483.
- [30] Stergios Koutsoumpas, Hasan Sayginel, Mark Webster, and Dan E. Browne, *Automorphism ensemble decoding of quantum LDPC codes*, [arXiv:2503.01738](https://arxiv.org/abs/2503.01738).
- [31] Robert J. McEliece, David J. C. MacKay, and Jung-Fu Cheng, *Turbo decoding as an instance of pearl’s “belief propagation” algorithm*, *IEEE J. Sel. Areas Commun.* **16**, 140 (1998).
- [32] Josias Old and Manuel Risper, *Generalized belief propagation algorithms for decoding of surface codes*, *Quantum* **7**, 1037 (2023).
- [33] Joschka Roffe, David R. White, Simon Burton, and Earl Campbell, *Decoding across the quantum low-density parity-check code landscape*, *Phys. Rev. Res.* **2**, 043423 (2020).
- [34] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss, *Understanding belief propagation and its generalizations*, in *Exploring Artificial Intelligence in the New Millennium* (Morgan Kaufmann, 2003), pp. 239–269, https://people.csail.mit.edu/billf/publications/Understanding_Belief_Propagation.pdf.
- [35] Martin J. Wainwright and Michael I. Jordan, *Graphical models, exponential families, and variational inference*, *Found. Trends Mach. Learn.* **1**, 1 (2008).
- [36] Gian Giacomo Guerreschi and Vadim N. Smelyanskiy, *Quantum approximate optimization with tensor network classical simulations*, *Phys. Rev. A* **101**, 022310 (2020).
- [37] Sergey Bravyi, Dan Browne, Padraic Calpin, Earl Campbell, David Gosset, and Robert Koenig, *Simulation of quantum circuits by low-rank stabilizer decompositions*, *Quantum* **3**, 181 (2019).
- [38] Joseph Tindall and Matt Fishman, *Gauging tensor networks with belief propagation*, *SciPost Phys.* **15**, 222 (2023).
- [39] Roy Alkabetz and Itai Arad, *Tensor networks contraction and the belief propagation algorithm*, *Phys. Rev. Res.* **3**, 023073 (2021).
- [40] Chu Guo, Dario Poletti, and Itai Arad, *Block belief propagation algorithm for two-dimensional tensor networks*, *Phys. Rev. B* **108**, 125111 (2023).
- [41] Aviad Kaufmann and Itai Arad, *A blockbp decoder for the surface code*, *Phys. Rev. Res.* **7**, 043025 (2025).
- [42] David Poulin and Ersen Bilgin, *Belief propagation algorithm for computing correlation functions in finite-temperature quantum many-body systems on loopy graphs*, *Phys. Rev. A* **77**, 052318 (2008).
- [43] David Poulin and Yeojin Chung, *On the iterative decoding of sparse quantum codes*, *Quantum Inf. Comput.* **8**, 986 (2008).
- [44] Subhayan Sahu and Brian Swingle, *Efficient tensor network simulation of quantum many-body physics on sparse graphs*, [arXiv:2206.04701](https://arxiv.org/abs/2206.04701).

- [45] Matthew B. Hastings, *Quantum belief propagation: An algorithm for thermal quantum systems*, *Phys. Rev. B* **76**, 201102 (2007).
- [46] Hao Chen and Thomas Barthel, *Tensor network states with low-rank tensors*, *arXiv:2205.15296*.
- [47] Yijia Wang, Yuwen Ebony Zhang, Feng Pan, and Pan Zhang, *Tensor network message passing*, *Phys. Rev. Lett.* **132**, 117401 (2024).
- [48] Joseph Tindall, Antonio Mello, Matt Fishman, Miles Stoudenmire, and Dries Sels, *Dynamics of disordered quantum systems with two- and three-dimensional tensor networks*, *Science* **392**, 868 (2026).
- [49] Manuel S. Rudolph and Joseph Tindall, *Simulating and sampling from quantum circuits with 2D tensor networks*, *arXiv:2507.11424*.
- [50] Glen Evenbly, Nicola Pancotti, Ashley Milsted, Johnnie Gray, and Garnet Kin-Lic Chan, *Loop series expansions for tensor networks*, *Phys. Rev. Res.* **8**, 013245 (2026).
- [51] Gunhee Park, Johnnie Gray, and Garnet Kin-Lic Chan, *Simulating quantum dynamics in two-dimensional lattices with tensor network influence functional belief propagation*, *Phys. Rev. B* **112**, 174310 (2025).
- [52] Michael Chertkov and Vladimir Y. Chernyak, *Loop calculus in statistical physics and information science*, *Phys. Rev. E* **73**, 065102 (2006).
- [53] Michael Chertkov and Vladimir Y. Chernyak, *Loop series for discrete statistical models on graphs*, *J. Stat. Mech.* (2006) P06009.
- [54] Michael Chertkov, Vicenc Gomez, and Hilbert Kappen, *Approximate inference on planar graphs using loop calculus and belief propagation*, *J. Mach. Learn. Res.* **11**, 1273 (2010).
- [55] Giorgio Parisi and František Štanina, *Loop expansion around the Bethe–Peierls approximation for lattice models*, *J. Stat. Mech.* (2006) L02003.
- [56] Andrea Montanari and Tommaso Rizzo, *How to compute loop corrections to the Bethe approximation*, *J. Stat. Mech.* (2005) P10011.
- [57] Joris Mooij and Hilbert Kappen, *Sufficient conditions for convergence of loopy belief propagation*, *arXiv:1207.1405*.
- [58] Vicenç Gómez, Joris M. Mooij, and Hilbert J. Kappen, *Truncating the loop series expansion for belief propagation*, *arXiv:cs/0612109*.
- [59] Abolfazl Ramezanpour and S. Moghimi-Araghi, *Statistical physics of loopy interactions: Independent-loop approximation and beyond*, *Phys. Rev. E* **92**, 032112 (2015).
- [60] Marcos Rigol, Tyler Bryant, and Rajiv R. P. Singh, *Numerical linked-cluster approach to quantum lattice models*, *Phys. Rev. Lett.* **97**, 187202 (2006).
- [61] Max Wellington, Andrew E. Gelfand, and Alexander T. Ihler, *A cluster-cumulant expansion at the fixed points of belief propagation*, *arXiv:1210.4916*.
- [62] Johnnie Gray, Gunhee Park, Glen Evenbly, Nicola Pancotti, and Garnet Kin Chan, *Tensor network loop cluster expansions for quantum many-body problems*, *arXiv:2510.05647*.
- [63] Note that no complex conjugation is taken here, in other words $I_{vw} = I_{wv}$.
- [64] For a graph (V, E) , any subset of edges $F \subseteq E$ defines the *edge-induced subgraph*, $G[F] = (V_F, F)$, where V_F is the set of vertices that are incident to edges in F .
- [65] Roman Kotecký and David Preiss, *Cluster expansion for abstract polymer models*, *Commun. Math. Phys.* **103**, 491 (1986).
- [66] See Supplementary Material at <http://link.aps.org/supplemental/10.1103/sdlx-d3tn> for [brief description], which includes Refs. [65, 77–79].
- [67] Sacha Friedli and Yvan Velenik, *Statistical Mechanics of Lattice Systems: A Concrete Mathematical Introduction* (Cambridge University Press, Cambridge, England, 2017).
- [68] Roberto Fernández and Aldo Procacci, *Cluster expansion for abstract polymer models. New bounds from an old approach*, *Commun. Math. Phys.* **274**, 123 (2007).
- [69] Christian Borgs, Jennifer Chayes, Jeff Kahn, and László Lovász, *Left and right convergence of graphs with bounded degree*, *Random Struct. Alg.* **42**, 1 (2013).
- [70] Technically, deduplication has a runtime of $O(n^2)$, but its constant factor is negligible.
- [71] Lars Onsager, *Crystal statistics. I. A two-dimensional model with an order-disorder transition*, *Phys. Rev.* **65**, 117 (1944).
- [72] Matthew Fishman, Steven White, and Edwin Miles Stoudenmire, *The ITensor software library for tensor network calculations*, *SciPost Phys. Codebases* 004 (2022).
- [73] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss, *Bethe free energy, Kikuchi approximations, and belief propagation algorithms*, *Adv. Neural Inf. Proces. Syst.* **13**, 24 (2001).
- [74] Rodney J. Baxter, *Exactly Solved Models in Statistical Mechanics* (Elsevier, 2016), https://shop.elsevier.com/books/exactly-solved-models-in-statistical-mechanics/baxter/978-0-12-083182-1?utm_source=chatgpt.com.
- [75] J. Ignacio Cirac Michael Lubasch and Mari-Carmen Banuls, *Algorithms for finite projected entangled pair states*, *Phys. Rev. B* **90**, 064425 (2014).
- [76] J. Ignacio Cirac Michael Lubasch and Mari-Carmen Banuls, *Unifying projected entangled pair state contractions*, *New J. Phys.* **16**, 033014 (2014).
- [77] Marc Mézard, *Andrea Montanari Information, Physics, and Computation* (Oxford University Press, 2009).
- [78] Leonardo Banchi, Jason Pereira, and Stefano Pirandola, *Generalization in quantum machine learning from few training data*, *Nat. Commun.* **12**, 6961 (2021).
- [79] Siddhant Midha, Yifan F. Zhang *et al.* (in preparation).