

The family of explicit belief propagation algorithms for arbitrary networks with loops

Pedro Hack^{*†}

October 7, 2026

Abstract

Although belief propagation (BP) is a useful inference tool in several network-based contexts given its speed, its precision deteriorates when encountering loops. Generalized belief propagation (GBP) was introduced around two decades ago to mitigate this issue. However, it did not become widely used since it did not provide an explicit algorithm applicable to any network. This only changed recently, with the introduction of the first explicit and general version of GBP for networks, the CN-method, which remained until now the only algorithm with such properties. Here, we provide a family of algorithms with these characteristics. We begin by considering the case where the network only has short loops. In this scenario, we improve on the CN-method by developing a algorithm with optimal time complexity, the NIB-method. We then consider networks that also have long loops. In this context, we show that the NIB-method provides advantages compared to the CN-method: it is faster for the networks where the CN-method is expected to work well, and it is applicable to more networks, i.e. highly dense networks. In general, both the CN-method and the family algorithms introduced here become non-equivalent approximations, and they can provide different levels of accuracy when applied to the same network, as we show numerically using two artificial networks. This discrepancy can have important practical implications, as we discuss in the context of decoding and error correction.

1 Introduction

There is a growing interest in the use of generalized message-passing schemes as inference tools [6, 11, 16, 20, 2, 19]. In fact, several disciplines rely on a graph or **network** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, that is, a set of points or nodes $\mathcal{V}$ together with a set of edges $\mathcal{E}$ between pairs of points, in order to study the collective behaviour of a set of individual entities (associated with the points) that interact in a pairwise fashion (associated with the edges) with each other. Networks, together with their generalization to higher order interactions among objects (i.e. **graphical models**) are used, for instance, in the study of disease spread, condensed matter physics, and both classical and quantum error correction [13, 3, 18, 7].

Given the large number of interactions in several systems of interest, extracting information is a challenging task. One of the most prominent algorithms to address this problem is BP [12, 14]. While BP is computationally non-expensive, its performance decreases when facing loops. Among the proposals to cope with this, generalized BP [23, 21] is based on the idea of computing correlations precisely within certain regions and then communicating this information to nearby regions in a similar fashion to how BP operates, achieving information transmission at a global scale by successively applying local communications which change, at each time step, according to the information each region received at the previous step. The lack of explicit constructions in generalized BP, together with the need for non-trivial preprocessing, led to the development of the **Cantwell-Newmann method** or **CN-method** for networks [11, 6], which continues to be the state-of-the-art approach [13, 17].

The CN-method introduces a set of neighborhoods $\{N_i\}_{i \in \mathcal{V}}$, one per node in the network, which are explicitly constructed in order to take care of the **short** loops around each node. Given two nodes which belong to a short loop $i, j \in \mathcal{V}$, the algorithm sends a message from i to j , and one from j to i , which essentially communicate the non-overlapping information from N_i to N_j and vice versa. The computation of such non-overlapping information is performed, however, in an inefficient way in the CN-method. By this we mean that the neighborhoods include uncorrelated pairs of nodes and, since such a

^{*}German Aerospace Center, Institute for Communication and Navigation, Wessling, Germany

[†]Technical University of Munich, CIT, Department of Computer Science, Garching, Germany

lack of correlation is not taken into account when sending messages, the complexity is unnecessarily large. This is particularly harmful in an exact implementation of the CN-method, since the complexity grows exponentially with the number of nodes. (In order to compare the methods, we focus here on the exact implementation of the CN-method. We make some remarks regarding approximate implementations in Section 9.) We can draw a direct connection between this issue and the introduction of standard BP, which was developed precisely by realizing that, given the simple correlations in a tree network, one can deal with several pairs of variables separately (hence, use less computational resources) without reducing the accuracy [15, 12].

1.1 Contribution

Our main contributions are the following:

- In the context of networks with only short loops, we develop an **optimal** message-passing algorithm, the **NIB-method** (Section 5).
- In the context of networks which also have long loops, we introduce the general (approximate) NIB-method (Section 6) and we interpolate between the NIB- and CN-methods, obtaining a rich family of explicit generalized belief propagation methods that can be applied to arbitrary networks (Section 7).

2 Networks

For our purposes here, a **network** is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each node $i \in \mathcal{V}$ is associated to a random variable X_i that takes values in a finite set X (which we assume to be independent of i for simplicity) and, given $i, j \in \mathcal{V}$, each edge $e = (i, j) \in \mathcal{E}$ is associated to a non-negative function $f_{i,j} : X \times X \rightarrow \mathbb{R}_{\geq 0}$ or **factor**. For simplicity, we assume the network to be **connected**¹. An instance of a network can be found in Figure 1a.

We think of the network as an encoding of some probability distribution that we are interested in

$$p(x_1, \dots, x_{|\mathcal{V}|}) = \frac{1}{Z} \prod_{(i,j) \in \mathcal{G}} f_{i,j}(x_i, x_j), \quad (1)$$

where

$$Z \equiv \sum_{x_1, \dots, x_{|\mathcal{V}|}} \prod_{(i,j) \in \mathcal{G}} f_{i,j}(x_i, x_j)$$

is a normalization constant or **partition function**².

In applications, networks may correspond to the probabilities of different spin configurations [11] or to the constraints satisfaction of variable configurations [12], like the compatibility of codewords with some observed syndrome.

In order to profit from statistical mechanics' tools, it is useful to associate to each function $f_{i,j}$ some energy $E_{i,j} \equiv -\log f_{i,j}$ ³. We can then think of (1) as a Boltzmann distribution

$$p(x_1, \dots, x_{|\mathcal{V}|}) = \frac{1}{Z} \prod_{(i,j) \in \mathcal{G}} e^{-E_{i,j}(x_i, x_j)}.$$

We will denote by $\mathcal{NN}_i$ the nearest neighbors of $i \in \mathcal{V}$, that is, the set of nodes $j \in \mathcal{V}$ that are connected to i by some edge $e = (i, j)$, and use the compact trace notation $\text{tr}(\cdot)$ ($\text{tr}_{\setminus A}(\cdot)$) to denote the sum over all variables (except those in A) that involve at least two functions within the parenthesis. We will refer to the number of nearest neighbors $|\mathcal{NN}_i|$ of a node $i \in \mathcal{V}$ as the **degree** of i .

¹Otherwise, we deal with its connected components individually.

²In statistical mechanics [11], it is customary to introduce networks by distinguishing between **interactions**, which correspond to the $f_{i,j}$ functions above, and **external potentials**, which are non-negative functions $f_i : X \rightarrow \mathbb{R}_{\geq 0}$ associated to each $i \in \mathcal{V}$. In our simplified picture, one can assume the external potentials have been absorbed into interactions with nearest neighbors.

³In practice, the case $f_{i,j}(x, y) = 0$ for some $x, y \in X$ will not be problematic, since we will assume in our definitions that $0 \log 0 \equiv 0$.

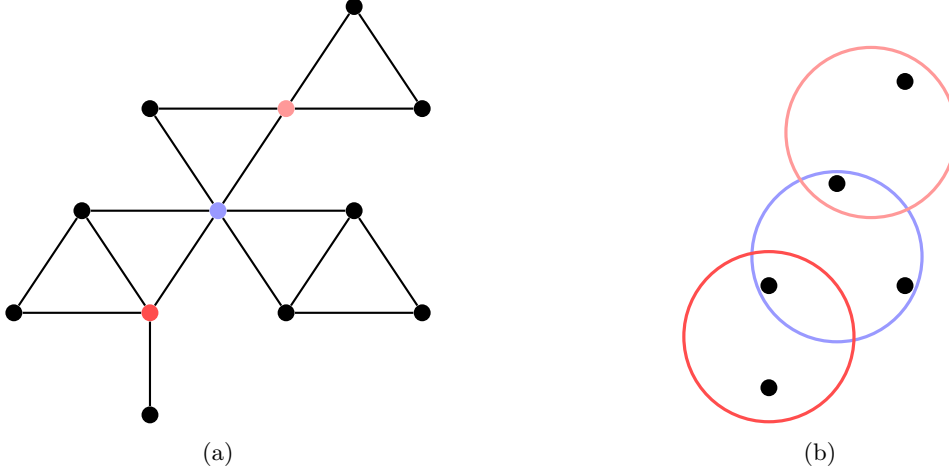


Figure 1: A network instance $\mathcal{G}_0$ (a), with nodes associated to variables and edges to interactions between pairs of variables. For $\mathcal{G}_0$, the NIB-method can be used with three pivots, represented by colored vertices. The NIB-associated-hypernetwork $\mathcal{G}_0/\sim$ (b), with one node for each equivalence class in the intersection quotient set $\cap/\sim$ and one hyperedge for each pivot in $\mathcal{G}_0$. The hyperedges encircle the nodes in the hypernetwork that they include and are colored like the pivot in $\mathcal{G}_0$ that they are associated to.

3 Standard BP on networks: Network BP

Standard BP is usually defined in the context of general graphical models and, as such, it includes two sorts of messages [12]: from factors to variables and from variables to factors. Given its restriction on factors, we can reduce this to a single type of messages between pairs of variable in the case of networks. More specifically, we define **network BP** via a family of one-dimensional messages between pairs of nearest neighbors

$$\begin{aligned} \{m_{i \rightarrow j}^{(\text{BP}, t)}\}_{i \in V, j \in \mathcal{NN}_i, t \geq 0}, \\ m_{i \rightarrow j}^{(\text{BP}, t)} : X \rightarrow \mathbb{R}_{\geq 0}. \end{aligned}$$

These messages are uniformly initialized, and updated, for $t \geq 0$, according to the following equation:

$$m_{i \rightarrow j}^{(\text{BP}, t+1)}(x_j) \propto \text{tr} \left(f_{i,j} \prod_{k \in \mathcal{NN}_i \setminus \{j\}} m_{k \rightarrow i}^{(\text{BP}, t)} \right),$$

where we omit a normalization constant that ensures the messages are probability distributions.

If we denote by

$$\{m_{i \rightarrow j}^{(\text{BP})}\}_{i \in V, j \in \mathcal{NN}_i}$$

the set of converged messages, then we can use the equations in Appendix A to infer quantities of interest like the partition function. These equations will be exact provided the network is a tree and only approximate if that is not the case.

Let us remark that this is not how standard BP is usually simplified in the network case [12]. The reason for this is that we are here restricted to some specific type of update equations, the **single variable BP algorithms** (see Appendix F), which we can then scale to take into account loops in the network. Since it does not possess this scaling property, this restriction does not include the usual BP reduction to networks. In any case, the time complexity only increases linearly with the size of the network, which is negligible compared to the exponential inefficiency in the CN-method that we intend to address here.

4 The CN-method

The CN-method [11, 6] goes beyond network BP by computing more correlations precisely. The number of correlations that are taken into account increases with a positive integer parameter $r \geq 0$, the **loop**

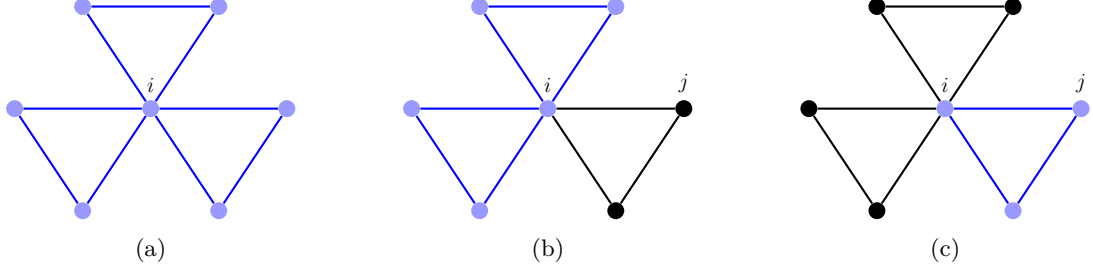


Figure 2: The sorts of neighborhoods we consider. In blue, the primary neighborhood $N_i^{(1)}$ (a), the difference neighborhood $N_{i \setminus j}^{(1)}$ (b), and the intersection neighborhood $N_{i \cap j}^{(1)}$ (c).

bound, that is fixed for each instance of the algorithm. For instance, if we take $r = 0$, then the CN-method takes into account the same correlations as standard BP.

For a given loop bound r , the CN-method is defined through two families of neighborhoods:

- the **primary** neighborhoods

$$\{N_i^{(r)}\}_{i \in \mathcal{V}}, \quad (2)$$

where $N_i^{(r)}$ consists of i together with its nearest neighbors $\mathcal{N}_i$ and the edges joining it to them, plus both edges and nodes along paths that join two nearest neighbors of i ;

- and the family of differences between these neighborhoods

$$\{N_{i \setminus j}^{(r)}\}_{i \in \mathcal{V}, j \in N_i^{(r)} \setminus \{i\}}, \quad (3)$$

where $N_{i \setminus j}^{(r)}$ consists of node i together with all the edges in $N_i^{(r)}$ which are not in $N_j^{(r)}$, and the nodes at the endpoints of such edges.

We will omit the superscript r in the following whenever it is clear.

The **CN-method**, or **r -CN-method** to be more specific, is defined via a family of one-dimensional messages between pairs of nodes that belong to each other's primary neighborhood

$$\begin{aligned} &\{m_{i \rightarrow j}^{(\text{CN}, t)}\}_{i \in \mathcal{V}, j \in N_i \setminus \{i\}, t \geq 0}, \\ &m_{i \rightarrow j}^{(\text{CN}, t)} : X \rightarrow \mathbb{R}_{\geq 0}. \end{aligned}$$

These messages are uniformly initialized, and updated, for $t \geq 0$, according to the following equation:

$$m_{i \rightarrow j}^{(\text{CN}, t+1)}(x_i) \propto \text{tr}_{x_i} \left(S_{i \setminus j} \prod_{k \in N_{i \setminus j} \setminus \{i\}} m_{k \rightarrow i}^{(\text{CN}, t)} \right), \quad (4)$$

where $S_{i \setminus j} \equiv \prod_{(k, q) \in N_{i \setminus j}} f_{k, q}$ stands for the product of the factors within the neighborhood difference $N_{i \setminus j}$, and we omit a normalization constant that ensures the messages are probability distributions.

If we denote by

$$\{m_{i \rightarrow j}^{(\text{CN})}\}_{i \in \mathcal{V}, j \in N_i}$$

the set of converged messages, then we can use the equations in Appendix B to infer quantities of interest like the partition function. These equations will be exact provided, for each $i \in \mathcal{V}$, all loops around i are contained within $N_i^{(r)}$. If that is the case, we say that the loop bound r is **fulfilled**, and that the network $\mathcal{G}$ has **r -bounded** loops. (**r -unbounded** loops if that is not the case.) Moreover, provided r is the smallest loop bound that is fulfilled, we say r is **sharp**. If r is not fulfilled, the inference equations in Appendix B are only approximate.

For a given network $\mathcal{G}_0$, we say the CN-method is **unfeasible** provided the time complexity of the update (4) or inference (Appendix B) equations is too large to be directly computed.

4.1 Computational resources in the CN-method

Intuitively, a generalized BP method should only be unfeasible provided the variables in the network are correlated too strongly (i.e. the network has several loops), and the method is using computational power in order to take such correlations into account. However, the suboptimal use of computational resources can **artificially** turn the CN-method unfeasible⁴. The best way to illustrate this is by showing that the CN-method is unfeasible for some networks with minimal variable correlations, that is, trees:

Claim 1 (0-CN vs. network BP). *The time complexity of the 0-CN-method grows exponentially with the maximal node degree of the network without providing any accuracy advantage with respect to network BP.*

Note that Claim 1, whose proof we delay until Appendix C, does not only address tree networks but any network that is treated as a tree by the CN-method. Intuitively, the claim makes sense since the 0-CN-method does not take into account more correlations than network BP. As a result of the claim, the CN-method is unfeasible on a tree network provided there is some node whose **degree** is too large. As in the case of network BP, the nodes' degree alone (that is, without any reference to loops or correlations) will not constitute a constraint for the exact applicability of the NIB-method, which we introduce in the next section.

5 An optimal algorithm: The NIB-method with r -bounded loops

We introduced the CN-method in Section 4 without distinguishing between the r -bounded and r -unbounded loops cases. In order to highlight the properties of the NIB-method, we differentiate between these cases when introducing it, addressing first the bounded case in this section, and dealing with the unbounded case in Section 6 (where we also remark that they could have been introduced simultaneously as in the CN case). Throughout this section, we assume some loop bound r_0 is fulfilled.

5.1 Intuitive picture

Before formally introducing the NIB-method and its properties when the loop bound is fulfilled, let us make an intuitive summary:

- The NIB-method begins by considering the intersection of pairs of regions used in the CN-method.
- Provided the loop bound is fulfilled, the intersection regions have a couple of useful properties:
 - There can be no loops that pass outside of an intersection region and then reenter it at a different node.
 - Given some intersection region A , and the intersection region B of any two different nodes that belong to A , we have that $A = B$. Intuitively, this is the case since, otherwise, say if there was some edge in B that does not belong to A , this edge would belong to a couple of loops, one around each of the points used to define B . We could use these loops in order to construct a loop that starts at a node in A , passes through edges outside A , and then reenters A at a different node. As argued before, such loops do not exist.
 - Given two distinct intersection regions $A \neq B$, we have that A and B share, at most, a single node (and hence a single edge). If they shared two nodes, and we take the intersection region associated to these two nodes C , we would have that $A = C$ and $B = C$ by the previous property. Hence, $A = B$, a contradiction.
- As a result, if we take one representative of each distinct intersection region, we get a tree decomposition of the whole network. We can then apply belief propagation to this tree to obtain exact results for the original network.

⁴Along this work, when referring to the CN-method, we mean the naive implementation of the exact message-passing scheme, as presented in [6, 11]. A more efficient implementation was considered [4], although it did not make a significant difference for the networks studied in [6].

5.2 The neighborhoods intersections

The NIB-method is defined through a subfamily of the intersections of the CN primary neighborhoods (2):

- The **neighborhoods intersections**

$$\begin{aligned} &\{N_{i \cap j}\}_{i \in \mathcal{V}, j \in N_i \setminus \{i\}}, \text{ where} \\ &N_{i \cap j} \equiv N_i \cap N_j \end{aligned} \quad (5)$$

and $\cap$ stands for the usual set intersection operator ⁵.

Intuitively, $N_{i \cap j}$ is the minimal set that includes all correlations between i and j . We will show its minimality later on. For now, it suffices to notice that it is constructed by taking into account the loops that include both i and j , plus paths that, although maybe not part of such a loop, cannot be disregarded without loosing exactness. (See Claim 8 in Appendix D for a characterization of $N_{i \cap j}$.)

The neighborhoods intersections fulfill an important property, which we refer to as the **equivalence class condition**:

Claim 2 (The equivalence class condition). *If the loop bound $r_0 \geq 0$ is fulfilled, $i \in \mathcal{V}$, and $j \in \mathcal{V} \cap (N_i^{(r_0)} \setminus \{i\})$, then*

$$N_{i \cap j}^{(r_0)} = N_{k \cap q}^{(r_0)} \quad (6)$$

for all $k, q \in \mathcal{V} \cap N_{i \cap j}^{(r_0)}$, $k \neq q$.

We prove Claim 2 in Appendix D, where we also give a counterexample to show that the condition on the loop bound r_0 cannot be avoided ⁶.

Claim 2 is useful since it allows us to split the family of neighborhoods intersections (hence the network) into equivalence classes such that two different classes either have no overlap or overlap on a single variable. The equivalence relation $\sim$ that we consider on the neighborhoods intersections is the following:

$$N_{i \cap j} \sim N_{k \cap q} \text{ if and only if } N_{i \cap j} = N_{k \cap q}.$$

This equivalence relation is associated to the **intersection quotient set**

$$\cap / \sim \equiv \{\overline{i \cap j}\}_{i \in \mathcal{V}, j \in N_i \setminus \{i\}},$$

where $\overline{i \cap j}$ denotes the equivalence class of $N_{i \cap j}$. By reduction to the absurd ⁷, we can show that, for $\overline{i \cap j}, \overline{k \cap q} \in \cap / \sim$ with $\overline{i \cap j} \neq \overline{k \cap q}$,

$$\begin{aligned} &|\mathcal{V} \cap (\overline{i \cap j} \cap \overline{k \cap q})| \in \{0, 1\}, \text{ and} \\ &\mathcal{E} \cap (\overline{i \cap j} \cap \overline{k \cap q}) = \emptyset. \end{aligned} \quad (7)$$

We refer to the vertices that belong to at least two different equivalence classes in $\cap / \sim$, that is, those $s \in \mathcal{V}$ for which there exist $\overline{i \cap j} \neq \overline{k \cap q}$ such that $s \in \overline{i \cap j} \cap \overline{k \cap q}$, as **pivots**, and denote the set of pivots in $\mathcal{G}$ by $\text{Piv}(\mathcal{G})$.

Pivots are useful in order to analyze an r -bounded network $\mathcal{G}$, since we can associate a hypernetwork to $\mathcal{G}$ through them:

- The **NIB-associated-hypernetwork**

$$\mathcal{G} / \sim \equiv (\cap / \sim, \{e_s\}_{s \in \text{Piv}(\mathcal{G})})$$

consists of the set of nodes $\cap / \sim$ together with one hyperedge e_s for each pivot $s \in \text{Piv}(\mathcal{G})$

$$e_s \equiv \{\overline{i \cap j} \in \cap / \sim \mid s \in \overline{i \cap j}\}.$$

⁵Note that the neighborhoods intersections do not include the intersection of **any** pair of primary neighborhoods, but only those between pairs of nodes that belong to each other's primary neighborhood. It is interesting to note that, while not using them as part of the message-passing scheme as the NIB-method does, Ref. [11] already considered the intersection neighborhoods as part of their inference equations.

⁶Claim 2 was already stated in the literature [11]. Nonetheless, we could not find a proof for it and, since it is central for the derivation of the NIB-method, we show it in Appendix D.

⁷If $|\mathcal{V} \cap (\overline{i \cap j} \cap \overline{k \cap q})| \geq 2$, then we take $r, s \in \mathcal{V} \cap (\overline{i \cap j} \cap \overline{k \cap q})$ with $r \neq s$ and we get, by Claim 2, $N_{i \cap j} = N_{r \cap s} = N_{k \cap q}$. This implies $\overline{i \cap j} = \overline{k \cap q}$, a contradiction. Moreover, if there exist $v_1, v_2 \in \mathcal{V}$, $v_1 \neq v_2$, such that $e = (v_1, v_2) \in \mathcal{E} \cap (\overline{i \cap j} \cap \overline{k \cap q})$, then, by definition of primary and intersections neighborhoods, $v_1, v_2 \in \mathcal{V} \cap (\overline{i \cap j} \cap \overline{k \cap q})$, contradicting the first statement.

The fulfillment of the loop bound implies that $\mathcal{G}/\sim$ is a **hypertree**, which is the key property that guarantees the exactness of the bounded NIB-method. We show in Figure 1a a network with three pivots, and in Figure 1b its NIB-associated-hypernetwork. We will denote by $\text{diam}(\mathcal{G}/\sim)$ the **diameter** [12] of the NIB-associated-hypernetwork.

5.3 Message-passing scheme

The **bounded neighborhoods-intersections-based** method (**bounded NIB-method**), or **bounded r -NIB-method** to be specific, is defined via a family of one-dimensional messages from the equivalence classes in the intersection quotient set into their constituent nodes:

$$\left\{ m_{\overline{i \cap j} \rightarrow i}^{(t)} \right\}_{i \in V, j \in (N_i \setminus \{i\})/\sim, t \geq 0},$$

$$m_{\overline{i \cap j} \rightarrow i}^{(t)} : X \rightarrow \mathbb{R}_{\geq 0},$$

where, given $j, k \in N_i \setminus \{i\}$,

$$j \sim k \text{ if and only if } \overline{i \cap j} = \overline{i \cap k}.$$

These messages are uniformly initialized, and updated, for $t \geq 0$, according to the following equation:

$$m_{\overline{i \cap j} \rightarrow i}^{(t+1)}(x_i) \propto \text{tr}_{x_i} \left(S_{i \cap j} \prod_{k \in N_{i \cap j} \setminus \{i\}} \prod_{\overline{k \cap q} \neq \overline{i \cap j}} m_{\overline{k \cap q} \rightarrow k}^{(t)} \right), \quad (8)$$

where $S_{i \cap j} \equiv \prod_{(k,q) \in N_{i \cap j}} f_{k,q}$ is the product of the factors in $N_{i \cap j}$ and we omit a normalization constant which ensures that the messages are probability distributions⁸.

If we denote by

$$\{ m_{\overline{i \cap j} \rightarrow i} \}_{i \in V, j \in (N_i \setminus \{i\})/\sim}$$

the set of converged messages, then we can use the equations in Appendix E to exactly infer quantities of interest like the partition function. These equations are exact provided the loop bound is fulfilled.

Intuitively, the NIB-method exploits the fact that, provided the loop bound is fulfilled, the NIB-associated-hypernetwork $\mathcal{G}/\sim$ is a tree-like partition of $\mathcal{G}$, and uses the pivots in $\mathcal{G}$ to (exactly) transport the information in each equivalence class to the others via the BP updates in (8). We show in Figure 1a a network on which the NIB-method uses three pivots.

5.4 Comparison to other message-passing schemes

Since the NIB-method is exact provided the loop bound is fulfilled, we can compare it solely in terms of complexity to other exact message-passing schemes in this context, like the CN-method. We devote the rest of this section to such a comparison.

We begin by comparing the NIB-method to the CN-method. Provided the loop bound $r_0 \geq 0$ is sharp, then one can show that, for most networks $\mathcal{G}$, the r_0 -NIB-method retains the exactness of the r_0 -CN-method while reducing its complexity:

Claim 3 (Bounded NIB vs. bounded CN). *If r_0 is a sharp loop bound, then the r_0 -NIB-method has a smaller time complexity than the r_0 -CN-method if and only if there is no pair $i, j \in \mathcal{V}$ such that $N_{i \cap j}^{(r_0)} = \mathcal{G}$.*

We prove Claim 3 in Appendix F. The statement relies on a speed-up in the inference complexity. A similar statement can be made regarding the update complexity, as we show in Appendix F. (See Claim 9.)

As a rough estimate of the complexity advantage of the NIB-method over the CN-method, consider two simplified asymptotic scenarios:

- If the complexity of inference in the CN-method is much larger than that of updating (i.e. that of each update step times the diameter $\text{diam}(\mathcal{G}/\sim)$), then the NIB-method requires $\mathcal{O}(|X|^\alpha)$ less time, where

$$\alpha \equiv \max_{i \in \mathcal{V}} \left\{ |\mathcal{V}_{N_i}| \right\} - \max_{\overline{i \cap j} \in \mathcal{G}/\sim} \left\{ |\mathcal{V}_{N_{i \cap j}}| \right\}$$

⁸If $N_i \setminus N_{i \cap j} = \emptyset$, then it will become clear that we do not need to update $m_{\overline{i \cap j} \rightarrow i}^{(t)}$ and, hence, this message will remain a uniform distribution for all $t \geq 0$.

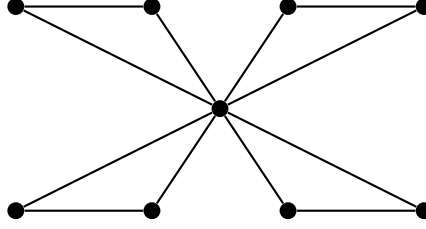


Figure 3: The single intersection n -triangle network with $n = 4$.

and $\mathcal{V}_A$ stands for the variables in $A \subseteq \mathcal{G}$. The improvement is exponential for almost any graph (see Claim 3).

- If the complexity of updating in the CN-method is much larger than that of inference, then the NIB-method requires $\mathcal{O}(|X|^\gamma)$ less time, where

$$\gamma \equiv \max_{i \in \mathcal{V}, j \in N_i \setminus \{i\}} \left\{ |\mathcal{V}_{N_{i \setminus j}}| \right\} - \max_{i \cap j \in \cap / \sim} \left\{ |\mathcal{V}_{N_{i \cap j}}| \right\}.$$

The improvement is exponential for several graphs (see Claim 9).

Given the advantage of using the NIB-method over the CN-method, one may ask whether further improvements are possible. This turns out to be impossible. More specifically, if we restrict ourselves to the class of **exact single variable BP algorithms** (see Appendix F for a definition), then the NIB-method cannot be improved upon:

Claim 4 (Bounded NIB optimality). *If r_0 is a sharp loop bound, then the r_0 -NIB-method is **optimal**, in the sense that, among all the exact single variable BP algorithms, the r_0 -NIB-method has minimal time complexity.*

We prove Claim 4 in Appendix F. As a particular case, this statement assures that network BP is optimal provided the network is loopless⁹.

The reduction in complexity that the NIB-method provides can have dramatic consequences regarding the feasibility of the method:

- If r_0 is a sharp loop bound, then there are networks where the r_0 -CN-method is **unfeasible** while the r_0 -NIB-method is not.

A simple instance where this holds is the **single intersection n -triangle** network, which consists of a central node c together with n triangles that include c and do not share any other node. In the 1-NIB-method, the time complexity of updating (and inference) is $\mathcal{O}(|X|^3)$ (independently of n), while that of the 1-CN-method is at least $\mathcal{O}(|X|^{2(n-1)+1})$, which becomes unfeasible for sufficiently large n . We show the single intersection n -triangle network with $n = 4$ in Figure 3. Recalling the inefficiencies of the CN-method we pointed out in Section 4.1, one can find loopless graphs where the transition from unfeasible CN to feasible NIB takes place. We will return to this in Section 6.3, where we mimic Section 4.1 but for the NIB-method.

Even if both methods are feasible, there are instances where the suboptimal use of computational resources in the CN-method allows us to go from innacurate CN results to accurate NIB results without spending more computational resources:

- Provided r_0 is a sharp loop bound and $r'_0 < r_0$, there are networks where the time complexity of the r_0 -NIB-method is bounded by that of the r'_0 -CN-method.

An instance where this happens with $r_0 = 3$ and $r'_0 = 1$ can be found in Figure 6b, with the same holding for any network with at least two **triangle-square** pairs (see Section 8 for a definition)¹⁰. While several other instances can be constructed using the intuition behind Figure 6b, characterizing all of them is not easy since the neighborhoods in each method are not related in a simple way when they use a different loop bounds.

⁹As discussed in Section 3, this is the case given our limited definition of the single variable BP algorithms. If we extend the definition, linear complexity reductions are possible.

¹⁰Numerically, we observe that the 1-CN-method achieves an error percentage $\sim 10^{-3}$ in the partition function, while the 3-NIB-method is exact.

6 The NIB-method on arbitrary networks

In this section, we do not assume some loop bound r is fulfilled and define the NIB-method in general.

6.1 Message-passing scheme

Since the loop bound may not be fulfilled, one cannot (see Appendix D) define the equivalence classes on which the bounded case relies. Because of this, we have to use the neighborhoods intersections (5) directly in order to define the algorithm. We will also have to be careful with the intersections between sets in this family, since they may no longer fulfill the simple relations in Claim 2 and (7).

The **NIB-method**, or **r -NIB-method** to be specific, is defined via a family of one-dimensional messages that sends, to each intersection of primary neighborhoods $N_{i \cap j}$, (part of) the information in $N_{k \cap q}$, the intersection of the primary neighborhoods of its constituent nodes $k \in N_{i \cap j}$ with other nodes in their primary neighborhoods $q \in N_k$:

$$\left\{ m_{k \cap q \rightarrow i \cap j}^{(t)} \right\}_{i \in V, j \in N_i \setminus \{i\}, k \in N_{i \cap j}, q \in N_k, t \geq 0},$$

$$m_{k \cap q \rightarrow i \cap j}^{(t)} : X \rightarrow \mathbb{R}_{\geq 0},$$

with $m_{k \cap q \rightarrow i \cap j}^{(t)}$ depending on variable x_k . We only send **part** of the information to avoid unnecessary redundancies. This will become clear in the next paragraph.

The messages are uniformly initialized and, in order for the message updates to avoid redundancies as much as possible, we need a couple of definitions. First, we consider a set $\mathcal{P}_{i \cap j}$, which we define recursively as follows:

- we initialize it by including all the functions within $N_{i \cap j}$;
- at each following step, we pick some $k \in N_{i \cap j}$ and some $q \in N_k$ and we incorporate the functions within $N_{k \cap q}$ to $\mathcal{P}_{i \cap j}$ ¹¹.

Given this set, we can define a map

$$\overline{\mathcal{P}_{i \cap j}} : \{N_{k \cap q}\}_{k \in N_{i \cap j}, q \in N_k} \rightarrow 2^{\mathcal{E}},$$

where $2^{\mathcal{E}}$ is the power set of $\mathcal{E}$, such that, for each pair $k, q \in \mathcal{V}$ that appears in the construction of $\mathcal{P}_{i \cap j}$, $\overline{\mathcal{P}_{i \cap j}}(N_{k \cap q})$ stands for $\mathcal{P}_{i \cap j}$ at the step right before the pair $k, q \in \mathcal{V}$ is selected.

Given these definitions, the messages are updated, for $t \geq 0$, according to the following equation:

$$m_{k \cap q \rightarrow i \cap j}^{(t+1)}(x_k) \propto \text{tr}_{x_k} \left(S_{k \cap q \setminus \overline{\mathcal{P}_{i \cap j}}(N_{k \cap q})} \prod_{p_1 \in N_{k \cap q} \setminus \{k\}} \prod_{p_2 \in N_{p_1}} m_{p_1 \cap p_2 \rightarrow k \cap q}^{(t)} \right),$$

where we have omitted a normalization constant that ensures the messages are probability distributions, and $S_{k \cap q \setminus \overline{\mathcal{P}_{i \cap j}}(N_{k \cap q})}$ stands for the product of the factors within $N_{k \cap q}$ which are not in $\overline{\mathcal{P}_{i \cap j}}(N_{k \cap q})$.

If we denote by

$$\{m_{k \cap q \rightarrow i \cap j}\}_{i \in V, j \in N_i \setminus \{i\}, k \in N_{i \cap j}, q \in N_k}$$

the set of converged messages, then we can use the equations in Appendix E to approximately infer quantities of interest like the partition function.

While we chose to introduce the bounded case separately, there is no need to make this distinction. To see this note that, if the loop bound r_0 is actually fulfilled, then, for all $t \geq 0$,

$$m_{k \cap q \rightarrow i \cap j}^{(t)} \equiv m_{k \cap q \rightarrow k}^{(t)}$$

provided $\overline{k \cap q} \neq \overline{i \cap j}$ and the pair $k, q \in \mathcal{V}$ is the first representative of the equivalence class $\overline{k \cap q}$ that is selected in the definition of $\mathcal{P}_{i \cap j}$, while $m_{k \cap q \rightarrow i \cap j}^{(t)}$ remains uniform otherwise, and hence does not affect the results.

It should be emphasized that it is at the inference level where we try to overcome the difficulties that the unbounded case presents when compared with the bounded one: In the bounded case, the

¹¹The choice of the ordering in the definition of each $\mathcal{P}_{i \cap j}$ can affect the performance of the NIB-method. Good orderings can be found by inspection.

intersection regions form equivalence classes, with different classes only intersecting at single variables. As a result, avoiding overcounting and obtaining exact inference equations is relatively simple (see Appendix E.1). In the unbounded case, however, the intersection regions do not form equivalence classes and can intersect with each other in diverse ways. Our proposed solution is to disregard part of these intersections when performing message-passing, that is, the messages that any region $N_{i \cap j}$ receives provide the non-overlapping information that the other intersections have with it. As a result, when combining the different intersection regions at the inference level, we may have to compute some coefficients (see Appendix E.2) to avoid overcounting.

It is also important to note that the lack of equivalence classes means that we must compute more messages in the unbounded case than in the bounded one. More importantly, we must compute more messages in the (bounded and unbounded, respectively) NIB-method than in their CN-method counterparts. However, the NIB-method offers a much better trade-off between space and time complexity: The space complexity disadvantage when compared with the CN-method grows (at most) quadratically with the number of variables that are incorporated into the principal neighborhoods (either because the network or r_0 grows), while its time complexity advantage grows exponentially with them.

6.2 Comparison to the CN-method

For the same loop bound, the time complexity of the unbounded NIB-method will always be bounded by that of the CN-method. We can even profit from the CN inefficiencies to go to higher loop bounds which, while still not being fulfilled, improve on the accuracy of the CN-method:

- If r_0 is a sharp loop bound and $r_0'' < r_0' < r_0$, then there are networks where the r_0' -NIB-method is more accurate than the r_0'' -CN-method without increasing its complexity.

An instance of this would be the 0-CN- and 1-NIB-method for the network in Figure 6b. In fact, this can even happen for $r_0'' = r_0' < r_0$, as we have numerically observed for the same network, where 2-NIB-method is (slightly) more accurate than the 2-CN-method.

The target network for which the CN-method method was tailored and where it is expected to give accurate results are the so-called **locally dense and globally sparse** networks [11]. Since this notion was not rigorously defined before, we introduce the following definition:

- Given a couple of integers $r_0, r_1 \geq 0$ with $r_0 \ll r_1$, a network is (r_0, r_1) **locally dense and globally sparse (LDGS)** provided

$$N_i^{(r_0)} = N_i^{(r_0+r_1)} \subsetneq \mathcal{G}$$

for all $i \in \mathcal{V}$.

That is, the set of loops is divided into two categories, with the length of those in one category being quite small w.r.t. those in the other. In this scenario, for all $i \in \mathcal{V}$ and $j \in N_i \setminus \{i\}$, there exists some $\mathcal{V}_{i \setminus j} \subseteq \mathcal{V}$ such that $N_{i \setminus j} = \cup_{v \in \mathcal{V}_{i \setminus j}} N_{i \cap v}$, where $N_{i \cap v_1} \cap N_{i \cap v_2} = \{i\}$ for all $v_1, v_2 \in \mathcal{V}_{i \setminus j}$. As a result, we have that:

Claim 5 (Improvement for LDGS networks). *If a network is (r_0, r_1) locally dense and globally sparse, then the r_0 -NIB-method has a smaller time complexity and achieves the same accuracy as the r_0 -CN-method.*

The proof of Claim 5 is analogous to that of Claim 3¹². Intuitively, the NIB-method is also optimal for LDGS networks. However, extending Claim 4 to this context is not analytically possible. An instance network which is (r_0, n) locally dense and globally sparse with $r_0 = 1$, and for which we have numerically confirmed the agreement between the methods, is the periodic n -triangle chain (see Figure 6a) with sufficiently large n . We give more details in Section 8.

The NIB-method offers decompositions that are not explored by the CN-method. In the context of highly dense networks, this can have important consequences regarding the applicability of the CN-method:

Claim 6 (Applicability to highly dense networks). *There exist highly dense networks where the r_0 -NIB-method is applicable for a range of parameters $r_0 > 0$ while the r_0 -CN-method is only applicable with $r_0 = 0$.*

¹²In fact, a decomposition analogous to (25) holds for locally dense and globally sparse networks as well.

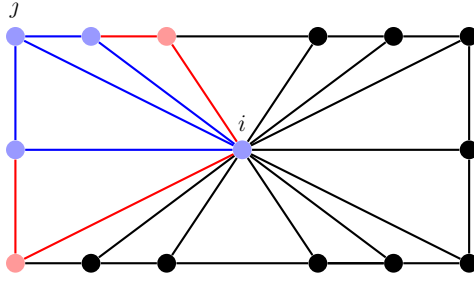


Figure 4: The n circumference network with $n = 14$. The CN-method with $r_0 = 1$ already takes the whole network as the primary neighborhood $N_i^{(1)}$, and is unfeasible for sufficiently large n . On the contrary, the NIB-method uses smaller neighborhoods, like $N_{i \cap j}^{(1)}$ (red) and $N_{i \cap j}^{(2)}$ (red and blue), and is hence feasible for some $r_0 > 0$.

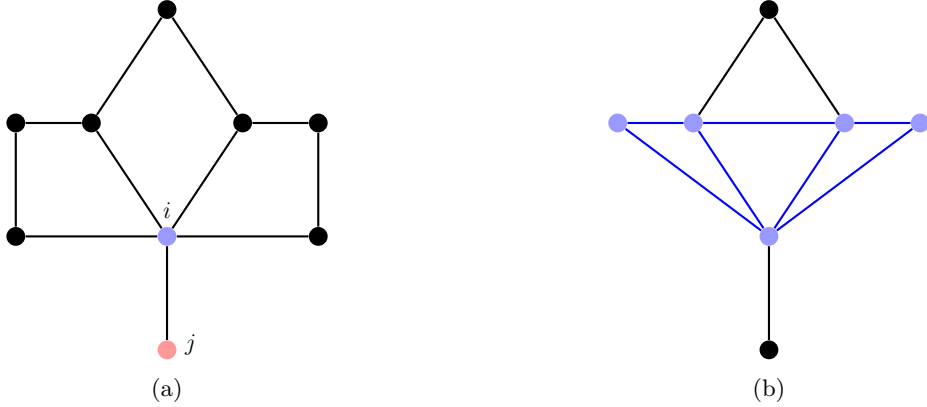


Figure 5: (a) The extra computational power that the CN-method deploys may be partially used to compute some correlations better provided some loop bound r_0 is not fulfilled. For instance, taking the blue node i and the red node j in the figure, the correlations computed within $N_{i \setminus j}$ in the 2-CN-method are not computed in the 2-NIB-method. (b) The CN-method is not always reproducible via the NIB-method. In this example, the blue region is used in the 1-CN-method and (in contrast with the bounded case) no equivalent set of regions is being used in any instance of the NIB-method.

For sufficiently large n , an example of such a network is the n **circumference** network, which consists of a circumference with n nodes on it and nearest neighbors connectivity, together with a node in the circumference's center which is connected to each of the other nodes. For sufficiently large n , the CN-method is unfeasible with $r_0 > 0$ and, hence, it does not improve on network BP. However, the NIB-method can be applied for several $r_0 > 0$ values. We show an example of the circumference network in Figure 4.

As a last comparison point, recall that, if some loop bound $r_0 \geq 0$ is fulfilled, the extra time complexity (compared to the r_0 -NIB-method) in the r_0 -CN-method is useless. This also applies to any (r_0, r_1) locally dense and globally sparse network. Nonetheless, this does not apply to all networks, with part of extra time complexity in the r_0 -CN-method sometimes allowing us to compute correlations between variables more accurately than in the r_0 -NIB-method. An instance network where this occurs with $r_0 = 2$ can be found in Figure 5a.

6.3 Computational resources in the NIB-method

As we discussed in Section 4.1, the inefficiencies in the CN-method become clear by comparison to network BP. In contrast, these inefficiencies are almost suppressed in the NIB-method:

Claim 7 (0-NIB vs. network BP). *The bounded 0-NIB-method is equivalent to standard BP. The unbounded 0-NIB-method provides the same accuracy as standard BP, with the complexity of its inference equations increasing by (at most) a factor of $|X|$.*

We prove Claim 7 in Appendix C. The statement ensures that the (bounded or unbounded) 0-NIB-method is essentially as applicable as network BP, in stark contrast to the 0-CN-method. More

specifically, the difference between network BP and the 0-NIB-method does not grow with the network, while the one with the 0-CN-method grows exponentially with the network density.

7 The family of explicit belief propagation algorithms

The boundary between the CN and NIB methods is soft, making it possible to interpolate between the two of them to get new message-passing schemes. As we show in Figure 5b, not all instances of the CN-method are reproducible via the NIB-method. Hence, interpolating between the methods becomes interesting in order to explore the **complexity-accuracy trade-off**.

By Claim 3, the interesting cases do not include networks where the loop bound is fulfilled. Moreover, by Claim 5, they also do not include locally dense and globally sparse networks. Lastly, they do not include any interpolation that involves the 0-CN-method. In any of these non-interesting cases we will end up with a method that provides the same accuracy as the NIB-method for a larger computational cost.

As argued in the previous paragraph, let us take a network $\mathcal{G}$ and an unsatisfied loop bound $r_0 \geq 1$ such that the r_0 -CN-method is not reproducible via the r_0 -NIB-method. By interpolation, and ordering them in non-decreasing complexity and (expected) non-decreasing accuracy, we obtain the following methods:

- the NIB-method, where the intersection neighborhoods (5) are used for both the update steps and inference;
- the **NIB-DIFF-method**, where the update steps use the intersection neighborhoods and the difference neighborhoods (3) are used for inference;
- the **CN-NIB-method**, where the update steps use the difference neighborhoods and the intersection neighborhoods are used for inference;
- the **NIB-CN-method**, where the update steps use the intersection neighborhoods and the primary neighborhoods (2) are used for inference;
- the CN-method, where the update steps use the difference neighborhoods and the primary neighborhoods are used for inference.

The CN- and NIB- methods have already been described in detail before. Moreover, the NIB-CN-method method is straightforward to envision from these two. We give some more details on the inference equations for both the NIB-DIFF and CN-NIB methods in Appendix G.

8 Numerical experiments

In order to support our previous statements numerically, we ought to show two properties:

- The NIB-method achieves the accuracy of the CN-method is locally dense and globally sparse networks.
- The different information flows that they follow can result in accuracy discrepancies.

To test the performance of the NIB-method, we compare it against the CN-method in two artificial networks with $|X| = 2$. We choose these networks, and the instances of the method we use for each of them, such that in one the NIB-method is an exact decomposition of the CN-method and, in the other one, they are different approximations of the network. The networks we consider are:

- The **periodic n -triangle chain** consists on n triangle networks $\{t_1, \dots, t_n\}$ such that each pair t_i, t_{i+1} share a single node for $i = 1, \dots, n - 1$ and t_n and t_1 also share a single node. Figure 6a shows the network for $n = 4$. Here, the 1-NIB-method is a decomposition of the 1-CN-method.
- The **single intersection n -triangle-square** consists of n pairs of triangles and squares such that each triangle is associated to a square, with which it shares two nodes, and all triangles share the node that they do not share with their associated squares. Figure 6b shows the network for $n = 4$. Here, the 2-NIB-method and the 2-CN-method are different approximations.

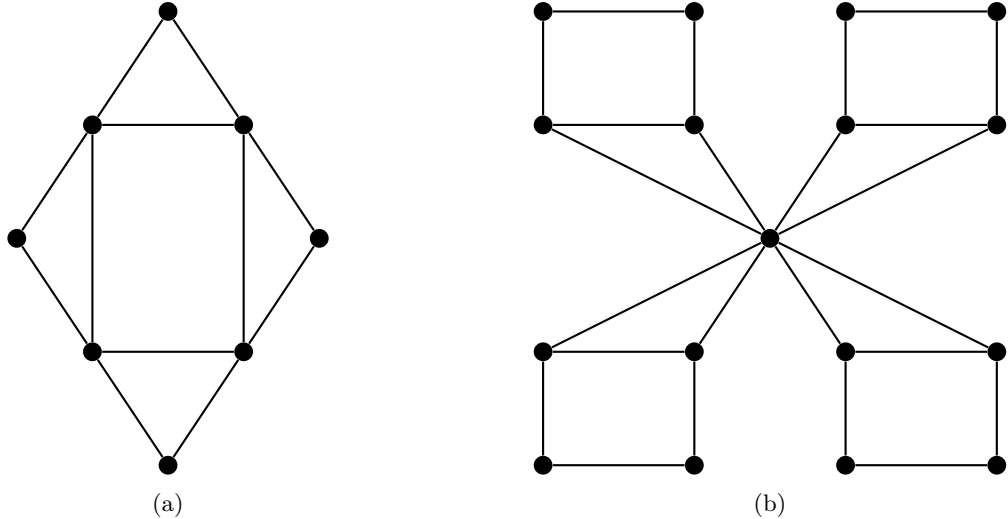


Figure 6: (a) The periodic n -triangle chain with $n = 4$. (b) The single intersection n -triangle-square with $n = 4$.

We have run 10^5 random instances of each network, sampling the values of the factors uniformly in the interval $(0, 1)$, and we have computed the mean percentage of error in the partition function using (a) the 1-CN and 1-NIB methods for the periodic n -triangle chain, and (b) the 2-CN and 2-NIB methods for the single intersection n -triangle-square. In both cases, we have found that, for different values of n between $n = 3$ and $n = 30$, and for a lower computational budget, the NIB-method achieves the same computational accuracy as its CN counterpart. In fact, for the single intersection n -triangle-square, and for small values of n , the NIB-method performs even slightly better than the CN-method, with the improvement vanishing as the network becomes larger. Figure ?? shows, for the periodic 30-triangle chain network, the variation of the partition function's mean percentage of error (in logarithmic scale) with the number of update steps.

We have also compared the CN and NIB methods using the **Ising model** on these networks, that is, we have used the same networks and we have taken, for each $e = (i, j) \in \mathcal{E}$, $f_{i,j}(x_i, x_j) \equiv \exp(-x_i x_j / T)$ instead of the random values from before, with $T > 0$ playing the role of the **temperature**. Figure ?? shows, for the single intersection 50-triangle-square network, the variation of the partition function's percentage of error (in logarithmic scale) with the temperature. Although we have observed an overall agreement between the NIB- and CN- methods, it is worth noting that for specific temperature values they can differ quite a bit. This happens because they are exchanging information in non-equivalent ways.

9 Applications

9.1 Decoding and error correction

The decoding problem arises in several applications where information storage or transmission is subject to errors. In order to account for this, one introduces redundant information such that, even if errors occur, one can in principle retrieve the original information. In practice, however, recovering this information with maximum likelihood can be computationally demanding. Hence, one needs to resort to fast algorithms, which may loose precision. That is, tackling the decoding problem consists of an exploration of the **complexity-accuracy trade-off**.

Since the effect of loops on performance is not well understood, it is not a priori clear what performance to expect from a given algorithm. However, the capabilities of each algorithm can be simulated. Hence, having explicit and generally applicable algorithms with potentially different performances like the ones introduced here is a key error correction tool. Some examples of the possible different performances are given in Section 8. In the decoding case, we are interested in a specific error probability, which we can interpret as a temperature value.

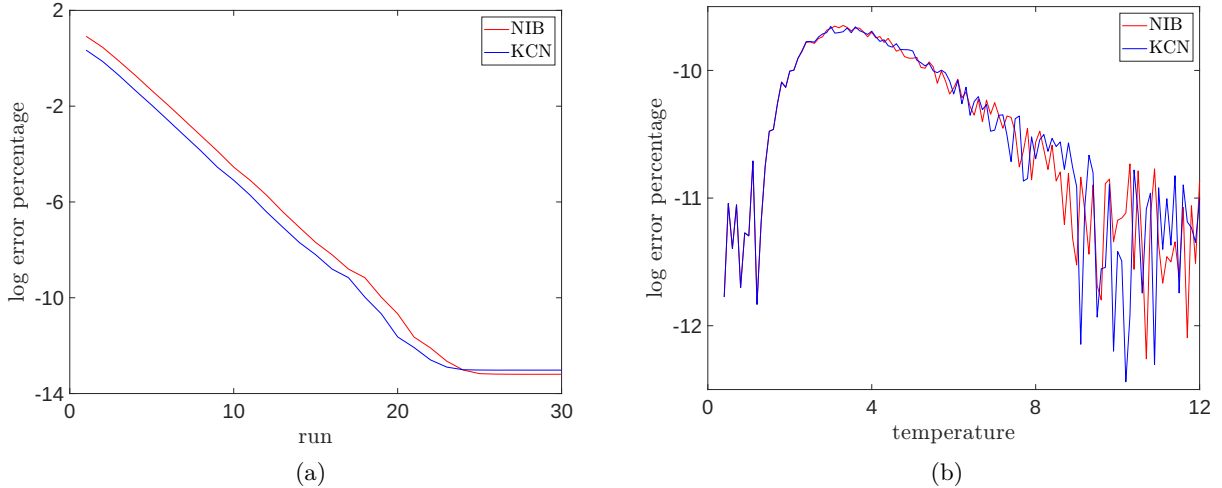


Figure 7: (a) Mean logarithmic percentage of error in the partition function Z as a function of the the number of update steps in the message-passing scheme for 10^5 random initializations of the periodic 30-triangle chain. We include, in red, the results for the 1-NIB-method and, in blue, those for the 1-CN-method. (b) Logarithmic percentage of error in the partition function Z as a function of the the temperature in the 50-triangle-square network Ising model. We include, in red, the results for the 2-NIB-method and, in blue, those for the 2-CN-method.

9.2 Approximate message passing: Combining BP and Monte Carlo sampling

Until now, we have considered the exact implementation of the message-passing equations for each method. Nonetheless, when the neighborhoods are too large and an exact implementation is not feasible, an approximate version of the CN-method which relies on **Monte Carlo sampling** in order to update the messages was also suggested in [11]. Such an approximate approach is also applicable for the NIB-method and the other algorithms introduced here. However, given the smaller neighborhoods in the family’s least expensive algorithm, the NIB-method, it provides advantages regarding Monte Carlo sampling when compared to the CN-method:

- The message passing equations can be exactly applied for larger values of r_0 in the NIB-method, thus postponing the second approximation layer that Monte Carlo sampling imposes.
- Monte Carlo sampling together with the NIB-method is more efficient in sampling large numbers of states of the network without taking very large samples of individual neighborhoods. That is, it provides an edge in what was considered [11] the most important advantage of combining Monte Carlo sampling and message passing when compared to traditional Monte Carlo methods.

These advantages occur for instance in the n circumference network (Figure 4) with sufficiently large n . Although one may argue that this is an artificial example, it actually happens in several real networks. For instance, one can take a model of a social network where nodes represent users and edges (i, j) represent that either user i follows user j or vice versa. In this context, several large instances of the n circumference network would appear, with the central node representing very popular users and the circumference nodes representing their non-popular followers.

10 Conclusion

Building on the CN-method, the only known explicit generalized belief propagation algorithm previously known, we have introduced here a family of algorithms with the same properties. This provides a toolbox that can be directly applied to any network, allowing us to explore the complexity-accuracy trade-off. While we have shown that some of the family members provide faster alternatives to the CN-method, they also provide different information flows depending on the network, resulting in different accuracy levels, as our simulations show. This may have important practical consequences, since the impact of loops in message-passing schemes is not well understood, and we cannot in general give preference to one method over another one in advance.

Regarding the time complexity, we can draw the following conclusions when comparing the fastest method in the family, the NIB-method, with the CN-method:

- If the loop bound is fulfilled, then:
 - for almost every network, the NIB-method is faster than the CN-method without losing accuracy;
 - no exact generalized belief propagation algorithm is faster than the NIB-method;
 - the NIB-method can be exactly applied to a larger family of networks than the CN-method.
- If the loop bound is not fulfilled, then:
 - the NIB-method provides, for a smaller time complexity, the same accuracy as the CN-method for the family of networks where the CN-method is expected to give accurate results, namely locally dense and globally sparse networks;
 - the NIB-method is directly applicable to networks where the high loop density imposes a second approximation layer in order for the CN-method to be used.

Regarding further directions, it would be interesting to assess the performance of the NIB-method beyond the artificial networks that we have considered here, specially in the context of decoding, where time complexity is essential. We will address the extension of our new explicit generalized belief propagation algorithms to other contexts in future work. In particular, our follow-up manuscript [9] deals with their extension to other applications in network theory [6] like percolation and the computation of matrix spectra. Another important extension is their adaptation to more general graphical models, which we recently tackled for the CN-method [10]. In this regard, the case of tensor networks is particularly interesting, since the use of message-passing schemes as inference tools has gained attention recently [1, 8, 20, 2]. Although we have been concerned with the CN neighborhoods here, it would be interesting to apply our proposal using a network version of the WZPZ-method [20, 10], where, essentially, neighborhoods larger than those in the CN-method are used and, crucially for us, the inefficiencies in the CN-method remain¹³. As a last remark, it should also be mentioned that Ref. [5] proposed another way of reducing the complexity of the CN-method, which essentially amounts to picking vertex-dependent loop bounds $\{r_v\}_{v \in \mathcal{V}}$. This approach can be naturally incorporated into our setup and, in fact, it will be required in less instances, since the size of the largest intersection neighborhoods can be considerably smaller than that of their primal counterparts.

Acknowledgements

I thank George T. Cantwell for useful feedback. This research is funded by the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus.

References

- [1] Roy Alkabetz and Itai Arad. Tensor networks contraction and the belief propagation algorithm. *Physical Review Research*, 3(2):023073, 2021.
- [2] Tomislav Begušić, Johnnie Gray, and Garnet Kin-Lic Chan. Fast and converged classical simulations of evidence for the utility of quantum computing before fault tolerance. *Science Advances*, 10(3):eadk4321, 2024.
- [3] Jacob C Bridgeman and Christopher T Chubb. Hand-waving and interpretive dance: an introductory course on tensor networks. *Journal of physics A: Mathematical and theoretical*, 50(22):223001, 2017.
- [4] George T Cantwell. Private Communication.
- [5] George T Cantwell, Alec Kirkley, and Filippo Radicchi. Heterogeneous message passing for heterogeneous networks. *Physical Review E*, 108(3):034310, 2023.

¹³See [10] for a discussion regarding the relation between the CN and WZPZ neighborhoods.

- [6] George T Cantwell and Mark EJ Newman. Message passing on networks with loops. *Proceedings of the National Academy of Sciences*, 116(47):23398–23403, 2019.
- [7] Andrew J Ferris and David Poulin. Tensor networks and quantum error correction. *Physical Review Letters*, 113(3):030501, 2014.
- [8] Chu Guo, Dario Poletti, and Itai Arad. Block belief propagation algorithm for two-dimensional tensor networks. *Physical Review B*, 108(12):125111, 2023.
- [9] Pedro Hack. Percolation and matrix spectrum through nib message passing. *arXiv preprint arXiv:2509.11730*, 2025.
- [10] Pedro Hack, Christian B Mendl, and Alexandru Paler. Belief propagation for general graphical models with loops. *arXiv preprint arXiv:2411.04957*, 2024.
- [11] Alec Kirkley, George T Cantwell, and MEJ Newman. Belief propagation for networks with loops. *Science Advances*, 7(17):eabf1211, 2021.
- [12] Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.
- [13] MEJ Newman. Message passing methods on complex networks. *Proceedings of the Royal Society A*, 479(2270):20220774, 2023.
- [14] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.
- [15] Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19(2):3, 2000.
- [16] Christophe Piveteau, Christopher T Chubb, and Joseph M Renes. Tensor network decoding beyond 2d. *arXiv preprint arXiv:2310.10722*, 2023.
- [17] Cheng Qian, Dan-Dan Zhao, Ming Zhong, Bo Zhang, Hao Peng, and Wei Wang. Message-passing approach for percolation on the networked system: A mini-review. *Europhysics Letters*, 2024.
- [18] Tom Richardson and Ruediger Urbanke. *Modern coding theory*. Cambridge university press, 2008.
- [19] Joseph Tindall, Matthew Fishman, E Miles Stoudenmire, and Dries Sels. Efficient tensor network simulation of IBM’s eagle kicked Ising experiment. *PRX Quantum*, 5(1):010308, 2024.
- [20] Yijia Wang, Yuwen Ebony Zhang, Feng Pan, and Pan Zhang. Tensor network message passing. *Physical Review Letters*, 132(11):117401, 2024.
- [21] Max Welling. On the choice of regions for generalized belief propagation. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 585–592. AUAI Press, 2004.
- [22] Robin J Wilson. *Introduction to graph theory*. Pearson Education India, 1979.
- [23] Jonathan S Yedidia, William Freeman, and Yair Weiss. Generalized belief propagation. *Advances in neural information processing systems*, 13, 2000.

A Inference in network BP

Given a tree network $\mathcal{G}$, the following inference equations are exact:

- For the marginals $p_i(\cdot)$ with $i \in \mathcal{V}$, we have that

$$p_i(x_i) \propto \prod_{k \in \mathcal{NN}_i} m_{k \rightarrow i}^{(\text{BP})}(x_i),$$

where we omit a normalization constant that ensures p_i is a probability distribution.

- For the internal energy U , we take the logarithm of the factors in $\mathcal{G}$ as energy terms and we get

$$U = - \sum_{(i,j) \in \mathcal{E}} \text{tr}(\log(f_{i,j}) p_{i,j}),$$

where

$$p_{i,j}(x_i, x_j) \equiv \frac{1}{Z_{i,j}} f_{i,j}(x_i, x_j) \prod_{k \in \mathcal{NN}_i \setminus \{j\}} m_{k \rightarrow i}^{(\text{BP})}(x_i) \times \prod_{q \in \mathcal{NN}_j \setminus \{i\}} m_{q \rightarrow j}^{(\text{BP})}(x_j),$$

and $Z_{i,j}$ a normalization constant that ensures $p_{i,j}$ is a probability distribution.

- For the Shannon entropy S , we have that

$$S = - \sum_{(i,j) \in \mathcal{E}} \text{tr}(p_{i,j} \log p_{i,j}) - \sum_{i \in \mathcal{V}} (1 - |\mathcal{NN}_i|) \text{tr}(p_i \log p_i). \quad (9)$$

- For the partition function Z , we have that

$$Z = \frac{\prod_{(i,j) \in \mathcal{E}} \text{tr}\left(f_{i,j} \prod_{k \in \mathcal{NN}_i \setminus \{j\}} m_{k \rightarrow i}^{(\text{BP})} \prod_{q \in \mathcal{NN}_j \setminus \{i\}} m_{q \rightarrow j}^{(\text{BP})}\right)}{\prod_{i \in \mathcal{V}} \text{tr}\left(\prod_{j \in \mathcal{NN}_i} m_{j \rightarrow i}^{(\text{BP})}\right)^{|\mathcal{NN}_i|-1}}. \quad (10)$$

As discussed in Section 3, we do not use here the typical simplification of standard BP on networks. Hence, we should derive its inference equations for completeness. For brevity, and since the other equations are not more involved to derive than this one, we only do so for the partition function Z :

Given a tree network, (10) holds since, taking $n_{i \rightarrow j}^{(\text{BP})} \equiv m_{i \rightarrow j}^{(\text{BP})} / M_{i \rightarrow j}^{(\text{BP})}$, with $M_{i \rightarrow j}^{(\text{BP})}$ being the product of the normalization constants that have been accumulated in $m_{i \rightarrow j}^{(\text{BP})}$ throughout the update process, we have:

$$\begin{aligned} & \frac{\prod_{(i,j) \in \mathcal{E}} \text{tr}\left(f_{i,j} \prod_{k \in \mathcal{NN}_i \setminus \{j\}} m_{k \rightarrow i}^{(\text{BP})} \prod_{q \in \mathcal{NN}_j \setminus \{i\}} m_{q \rightarrow j}^{(\text{BP})}\right)}{\prod_{i \in \mathcal{V}} \text{tr}\left(\prod_{j \in \mathcal{NN}_i} m_{j \rightarrow i}^{(\text{BP})}\right)^{|\mathcal{NN}_i|-1}} \\ &= \frac{\prod_{(i,j) \in \mathcal{E}} \text{tr}\left(f_{i,j} \prod_{k \in \mathcal{NN}_i \setminus \{j\}} n_{k \rightarrow i}^{(\text{BP})} \prod_{q \in \mathcal{NN}_j \setminus \{i\}} n_{q \rightarrow j}^{(\text{BP})}\right)}{\prod_{i \in \mathcal{V}} \text{tr}\left(\prod_{j \in \mathcal{NN}_i} n_{j \rightarrow i}^{(\text{BP})}\right)^{|\mathcal{NN}_i|-1}} \\ &= \frac{\prod_{(i,j) \in \mathcal{E}} Z}{\prod_{i \in \mathcal{V}} Z^{|\mathcal{NN}_i|-1}} = Z^{|\mathcal{E}| - (\sum_{i \in \mathcal{V}} (|\mathcal{NN}_i| - 1))} \\ &= Z^{|\mathcal{V}| - 1 - (2(|\mathcal{V}| - 1) - |\mathcal{V}|)} = Z, \end{aligned}$$

where we have cancelled the normalization constants $M_{i \rightarrow j}^{(\text{BP})}$ in the first equality since they appear exactly $|\mathcal{NN}_i| - 1$ times in the numerator and only once in the denominator, and we have used [22, Theorem 9.1] in the second to last.

When facing a network with loops, we can use the previous equations for approximate inference.

It is important to note that the essence of these equations is the substitution of the probability distribution associated to a network by another probability distribution composed of independent terms associated to some of its subnetworks. More specifically, network BP is based on the following exact decomposition for tree networks:

$$p(x_1, \dots, x_{|\mathcal{V}|}) \propto \prod_{(i,j) \in \mathcal{E}} \left(f_{i,j}(x_i, x_j) \prod_{k \in \mathcal{NN}_i \setminus \{j\}} m_{k \rightarrow i}^{(\text{BP})}(x_i) \prod_{q \in \mathcal{NN}_j \setminus \{i\}} m_{q \rightarrow j}^{(\text{BP})}(x_j) \right) \times \prod_{i \in \mathcal{V}} \left(\prod_{j \in \mathcal{NN}_i} m_{j \rightarrow i}^{(\text{BP})}(x_i) \right)^{1 - |\mathcal{NN}_i|}. \quad (11)$$

This decomposition is useful to obtain some of the inference equations from the rest. For instance, under the loopless assumption, we can use (11) to get another exact equation for the internal energy

$$U = - \sum_{(i,j) \in \mathcal{E}} \text{tr} \left(p_{i,j} \left(\log(f_{i,j}) + \sum_{k \in \mathcal{NN}_i \setminus \{j\}} \log(m_{k \rightarrow i}^{(\text{BP})}) + \sum_{q \in \mathcal{NN}_j \setminus \{i\}} \log(m_{q \rightarrow j}^{(\text{BP})}) \right) \right) - \sum_{i \in \mathcal{V}} (1 - |\mathcal{NN}_i|) \text{tr} \left(p_i \sum_{j \in \mathcal{NN}_i} \log(m_{j \rightarrow i}^{(\text{BP})}) \right), \quad (12)$$

which we can use together with (9) and the well-known equation

$$Z = e^{S-U} \quad (13)$$

to obtain (10).

Importantly, if the loopless assumption does not hold, all the inference equations are based on the same assumption, namely that the decomposition (11) holds, and we can think of them as being determined once the decomposition is fixed. This will be relevant when discussing inference in the CN-method in Appendix B.

B Inference in the CN-method

The inference equations for the CN-method were already provided when it was introduced [11] (see for instance [11, Eq. 11] for the marginals, [11, Eq. 14] for the internal energy, and [11, Eq. 22] for the Shannon entropy). We devote this section to making a couple of comments on them.

Our first comment concerns the inference of the partition function. Despite it being the central quantity of interest [11], no direct equation in terms of messages is derived for it. More specifically, in contrast to their equations for the internal energy of the Shannon entropy, the partition function is derived via (13). This is interesting when contrasted with the discussion regarding network decomposition in Appendix A. In particular, [11] deploys one sort of network decomposition when dealing with the internal energy and a different one when dealing with the Shannon entropy. However, we can use the composition from the Shannon entropy (the one for the internal energy is not general enough) to obtain an inference equation for the partition function:

$$Z = \frac{\prod_{((i,j)) \in \mathcal{G}} \text{tr} \left(S_{i \cap j} m_{j \rightarrow i}^{(\text{CN})} \prod_{k \in N_{i \cap j} \setminus \{j\}} m_{k \rightarrow j}^{(\text{CN})} \right)^{\frac{1}{\binom{|S_{i \cap j}|}{2}}}}{\prod_{(i,j) \in \mathcal{E}} \text{tr} \left(S_{i \cap j} m_{j \rightarrow i}^{(\text{CN})} \prod_{k \in N_{i \cap j} \setminus \{j\}} m_{k \rightarrow j}^{(\text{CN})} \right)^{-W_{i,j}} \prod_{i \in \mathcal{V}} \text{tr} \left(S_i \prod_{j \in N_i \setminus \{i\}} m_{j \rightarrow i}^{(\text{CN})} \right)^{-C_i}}, \quad (14)$$

where S_i stands for the functions within N_i , $((i,j))$ stands for the pairs $i, j \in \mathcal{G}$ such that $i \in N_j$ and, hence, $j \in N_i$, $|S_{i \cap j}|$ stands for the number of variables within $S_{i \cap j}$, and

$$W_{i,j} \equiv 1 - \sum_{((\ell,k)) \in \mathcal{G}} \frac{1}{\binom{|S_{\ell \cap k}|}{2}} \chi_{\{(i,j) \in N_{\ell,k}\}}, \quad C_i \equiv 1 - \sum_{j \in N_i} \left(\frac{1}{|S_{i \cap j}| - 1} \right) - \sum_{j \in \mathcal{NN}_i} W_{i,j}. \quad (15)$$

Note that, if $r = 0$, then (14) holds with $W_{i,j} = 0$ for all $(i,j) \in \mathcal{G}$. We will focus on this case in Appendix C.

Our second comment is a short clarification regarding the inference of the two-variable marginals $p_{i,j}$, which appear in the inference equation for the Shannon entropy in the CN-method. In general [12, 11], to get such a marginal for any couple of variables requires running a message-passing method (say the CN-method) once to estimate a single variable marginal $p_i(\cdot)$ and then running it again, once for each value $x_i \in X$, in order to obtain the single variable marginal $p_j^{x_i}$ conditioned on $X_i = x_i$. This process is computationally expensive. Nonetheless, in the CN-method we are only interested in $p_{i,j}$ for $(i,j) \in \mathcal{E}$ and, hence, we can estimate it via

$$p_{i,j}(x_i, x_j) \propto \frac{1}{2Z_i} \text{tr}_{\setminus(x_i, x_j)} \left(S_i \prod_{k \in N_i \setminus \{i\}} m_{k \rightarrow i}^{(\text{CN})} \right) + \frac{1}{2Z_j} \text{tr}_{\setminus(x_i, x_j)} \left(S_j \prod_{k \in N_j \setminus \{j\}} m_{k \rightarrow j}^{(\text{CN})} \right),$$

where Z_i and Z_j are normalization constants that ensure each of the summands is a probability distribution. Later on, we will use a similar equation for $p_{i,j}$ which is based on the intersection set $N_{i \cap j}$.

C Computational resources in the CN and NIB methods

C.1 Proof of Claim 1

By definition, for all $t \geq 0$, we have the following relations:

$$\begin{aligned} m_{j \rightarrow i}^{(\text{CN}, t)}(x_j) &\propto \prod_{k \in \mathcal{NN}_j \setminus \{i\}} m_{k \rightarrow j}^{(\text{BP}, t-1)}(x_j), \\ m_{j \rightarrow i}^{(\text{BP}, t)}(x_i) &\propto \sum_{x_j} f_{i,j}(x_i, x_j) m_{j \rightarrow i}^{(\text{CN}, t-1)}(x_j). \end{aligned} \tag{16}$$

These equations imply that, if one of the methods converges, then the other one does as well. Assume, hence, that both methods have converged, respectively, to the messages

$$\begin{aligned} \{m_{i \rightarrow j}^{(\text{BP})}\}_{i \in \mathcal{V}, j \in \mathcal{NN}_i, t \geq 0} \text{ and} \\ \{m_{i \rightarrow j}^{(\text{CN})}\}_{i \in \mathcal{V}, j \in \mathcal{NN}_i, t \geq 0}. \end{aligned}$$

We conclude by showing that they infer the same quantities:

Regarding marginals, we have that

$$\begin{aligned} p_i^{(\text{CN})}(x_i) &\propto \text{tr}_{\setminus x_i} \left(\prod_{j \in \mathcal{NN}_i} f_{i,j} m_{j \rightarrow i}^{(\text{CN})} \right) \\ &\propto \prod_{j \in \mathcal{NN}_i} \text{tr}_{\setminus x_i} \left(f_{i,j} m_{j \rightarrow i}^{(\text{CN})} \right) \\ &\propto \prod_{j \in \mathcal{NN}_i} m_{j \rightarrow i}^{(\text{BP})}(x_i) \\ &\propto p_i^{(\text{BP})}(x_i). \end{aligned}$$

This implies that $p_i^{(\text{CN})} = p_i^{(\text{BP})}$ by normalization.

Regarding internal energy, we have that

$$\begin{aligned}
U^{(\text{BP})} &= - \sum_{(i,j) \in \mathcal{E}} \text{tr} \left(\log f_{i,j} \frac{f_{i,j} \prod_{k \in \mathcal{NN}_i \setminus \{j\}} m_{k \rightarrow i}^{(\text{BP})} \prod_{q \in \mathcal{NN}_j \setminus \{i\}} m_{q \rightarrow j}^{(\text{BP})}}{\text{tr} \left(f_{i,j} \prod_{k \in \mathcal{NN}_i \setminus \{j\}} m_{k \rightarrow i}^{(\text{BP})} \prod_{q \in \mathcal{NN}_j \setminus \{i\}} m_{q \rightarrow j}^{(\text{BP})} \right)} \right) \\
&= - \sum_{(i,j) \in \mathcal{E}} \text{tr} \left(\log f_{i,j} \frac{f_{i,j} m_{i \rightarrow j}^{(\text{CN})} m_{j \rightarrow i}^{(\text{CN})}}{\text{tr} \left(f_{i,j} m_{i \rightarrow j}^{(\text{CN})} m_{j \rightarrow i}^{(\text{CN})} \right)} \right) \\
&= - \frac{1}{2} \sum_{i \in \mathcal{V}, j \in \mathcal{NN}_i} \text{tr} \left(\log f_{i,j} \frac{\prod_{j \in \mathcal{NN}_i} f_{i,j} m_{j \rightarrow i}^{(\text{CN})}}{\text{tr} \left(\prod_{j \in \mathcal{NN}_i} f_{i,j} m_{j \rightarrow i}^{(\text{CN})} \right)} \right) \\
&= U^{(\text{CN})},
\end{aligned}$$

where we have used (16) in the second equality, and the fact that the messages have converged in the third equality.

Regarding entropy, we have that

$$\begin{aligned}
S^{(\text{CN})} &= - \sum_{(i,j) \in \mathcal{E}} \text{tr} \left(\frac{f_{i,j} m_{i \rightarrow j}^{(\text{CN})} m_{j \rightarrow i}^{(\text{CN})}}{\text{tr} \left(f_{i,j} m_{i \rightarrow j}^{(\text{CN})} m_{j \rightarrow i}^{(\text{CN})} \right)} \log \left(\frac{f_{i,j} m_{i \rightarrow j}^{(\text{CN})} m_{j \rightarrow i}^{(\text{CN})}}{\text{tr} \left(f_{i,j} m_{i \rightarrow j}^{(\text{CN})} m_{j \rightarrow i}^{(\text{CN})} \right)} \right) \right) \\
&\quad - \sum_{i \in \mathcal{V}} (1 - |\mathcal{NN}_i|) \text{tr} \left(\frac{\text{tr}_{\setminus x_i} \left(\prod_{j \in \mathcal{NN}_i} f_{i,j} m_{j \rightarrow i}^{(\text{CN})} \right)}{\text{tr} \left(\prod_{j \in \mathcal{NN}_i} f_{i,j} m_{j \rightarrow i}^{(\text{CN})} \right)} \log \left(\frac{\text{tr}_{\setminus x_i} \left(\prod_{j \in \mathcal{NN}_i} f_{i,j} m_{j \rightarrow i}^{(\text{CN})} \right)}{\text{tr} \left(\prod_{j \in \mathcal{NN}_i} f_{i,j} m_{j \rightarrow i}^{(\text{CN})} \right)} \right) \right) \\
&= - \sum_{(i,j) \in \mathcal{E}} \text{tr} \left(\frac{f_{i,j} \prod_{k \in \mathcal{NN}_i \setminus \{j\}} m_{k \rightarrow i}^{(\text{BP})} \prod_{q \in \mathcal{NN}_j \setminus \{i\}} m_{q \rightarrow j}^{(\text{BP})}}{\text{tr} \left(f_{i,j} \prod_{k \in \mathcal{NN}_i \setminus \{j\}} m_{k \rightarrow i}^{(\text{BP})} \prod_{q \in \mathcal{NN}_j \setminus \{i\}} m_{q \rightarrow j}^{(\text{BP})} \right)} \times \right. \\
&\quad \left. \log \left(\frac{f_{i,j} \prod_{k \in \mathcal{NN}_i \setminus \{j\}} m_{k \rightarrow i}^{(\text{BP})} \prod_{q \in \mathcal{NN}_j \setminus \{i\}} m_{q \rightarrow j}^{(\text{BP})}}{\text{tr} \left(f_{i,j} \prod_{k \in \mathcal{NN}_i \setminus \{j\}} m_{k \rightarrow i}^{(\text{BP})} \prod_{q \in \mathcal{NN}_j \setminus \{i\}} m_{q \rightarrow j}^{(\text{BP})} \right)} \right) \right) \\
&\quad - \sum_{i \in \mathcal{V}} (1 - |\mathcal{NN}_i|) \text{tr} \left(\frac{\prod_{k \in \mathcal{NN}_i} m_{k \rightarrow i}^{(\text{BP})}}{\text{tr} \left(\prod_{k \in \mathcal{NN}_i} m_{k \rightarrow i}^{(\text{BP})} \right)} \log \left(\frac{\prod_{k \in \mathcal{NN}_i} m_{k \rightarrow i}^{(\text{BP})}}{\text{tr} \left(\prod_{k \in \mathcal{NN}_i} m_{k \rightarrow i}^{(\text{BP})} \right)} \right) \right) \\
&= S^{(\text{BP})},
\end{aligned}$$

where we have used (16) in the second equality.

Regarding the partition function, we can proceed along the lines of the entropy equality to get $Z^{(\text{CN})} = Z^{(\text{BP})}$, with $Z^{(\text{CN})}$ defined as in (14).

It is important to note that we have not required the network to be a tree at any step.

C.2 Proof of Claim 7

In the bounded case, we have that

$$m_{i \cap j \rightarrow i}^{(\text{NIB}, t)} \equiv m_{i \rightarrow j}^{(\text{BP}, t)}.$$

Moreover, the inference equations also coincide.

In the unbounded case, and provided there exists some $k \in \mathcal{NN}_j \setminus \{i\}$, we have that

$$m_{i \cap j \rightarrow j \cap k}^{(\text{NIB}, t)} \equiv m_{i \rightarrow j}^{(\text{BP}, t)}.$$

This means that network BP computes more messages than the unbounded 0-NIB-method, since $m_{i \rightarrow j}^{(\text{BP}, t)}$ has no NIB counterpart provided $\mathcal{NN}_j \setminus \{i\} = \emptyset$.

The relation among the messages implies that, if one of the methods converges, then the other one does as well. To convince ourselves of this, we simply note that the messages $m_{i \rightarrow j}^{(\text{BP}, t)}$ which are not computed in the unbounded 0-NIB-method do not affect the rest of the messages, since we have $\mathcal{NN}_j \setminus \{i\} = \emptyset$.

From the convergence assumption and the inference equations, it becomes clear that the inferred quantities are the same. Regarding time complexity, the methods only differ in the one associated to

inference, where the NIB-method requires, at most, a factor of $|X|$ more time. This increase takes place in the estimation of marginals, where the complexity rises from $|X|$ to $|X|^2$. Regarding other quantities, the complexity of the partition function doubles, while that of the internal energy stays the same.

D Properties of neighborhoods intersections

D.1 The equivalence class condition:

Proof of Claim 2

To show the result, it suffices to argue that, given $k \in N_{i \cap j}$, we have $N_{i \cap j} \subseteq N_k$. This is the case since, if this holds, it also does for $q \in N_{i \cap j}$, and we get that $N_{i \cap j} \subseteq N_{k \cap q}$. Since $i, j \in N_{k \cap q}$, we analogously get that $N_{k \cap q} \subseteq N_{i \cap j}$ and we obtain the result $N_{i \cap j} = N_{k \cap q}$.

To conclude, hence, we show that $N_{i \cap j} \subseteq N_k$ provided $k \in N_{i \cap j}$. Note first that it, if we show the result for edges $\mathcal{E} \cap N_{i \cap j} \subseteq N_k$, then the result holds for vertices $\mathcal{V} \cap N_{i \cap j} \subseteq N_k$ by definition.

We show in what follows that $e \in N_k$ for any edge $e \in \mathcal{E} \cap N_{i \cap j}$. In order to do so, let us first construct a path p_{i-e-j} that goes from i to j through e : We start by taking a cycle c_{i-e-i} from i to itself that passes through e . We also consider a path p_{j-e} that starts at j and finishes at e , and we take $r_0 \in \mathcal{V}$ the closest vertex to j in p_{j-e} such that $r_0 \in c_{i-e-i} \cap p_{j-e}$. (When referring to two elements x, y belonging to a path p , by **distance**, and hence by **closest**, with respect to p we refer to the number of edges and vertices that separate x from y along p .) We take

$$p_{i-e-j} \equiv p_{j-r_0} \cup p_{r_0-e-i},$$

where p_{j-r_0} is the path within p_{j-e} that joins j and r_0 , and p_{r_0-e-i} is the path within c_{i-e-i} that goes from r_0 to i through e . By construction, p_{i-e-j} is a path.

We now want to use p_{i-e-j} to form a cycle around k that contains e . To do so, note that p_{i-e-j} intersects with c_{k-j-k} and c_{k-i-k} , which are, respectively, cycles that include k and j , and k and i . We denote by $r_i \in \mathcal{V}$ (and analogously by $r_j \in \mathcal{V}$) the vertex between e and i (e and j) that is closest to e in p_{i-e-j} such that $r_i \in c_{k-j-k} \cup c_{k-i-k}$ ($r_j \in c_{k-j-k} \cup c_{k-i-k}$), and note that $r_i \neq r_j$.

We conclude by considering two cases:

- (I) We can take $c \in \{c_{k-j-k}, c_{k-i-k}\}$ such that $r_i, r_j \in c$. Then, we take

$$c_{k-e-k} \equiv p_{k-r_i} \cup p_{r_i-e-r_j} \cup p_{r_j-k},$$

where p_{k-r_i} and p_{r_j-k} are the subpaths of c that, respectively, join r_i and k , and r_j and k , and fulfill $p_{k-r_i} \cap p_{r_j-k} = \{k\}$, and $p_{r_i-e-r_j}$ the subpath of p_{i-e-j} that joins r_i and r_j . By construction, c_{k-e-k} is a cycle around k that includes e and, hence, $e \in N_k$.

- (II) If $r_i \in c_1$ and $r_j \in c_2$, with $c_1, c_2 \in \{c_{k-j-k}, c_{k-i-k}\}$ and $c_1 \neq c_2$, then, assuming w.l.o.g. that $r_i \in c_{k-i-k}$, we take

$$c_{k-e-k} \equiv p_{k-s} \cup p_{s-r_i} \cup p_{r_i-e-r_j} \cup p_{r_j-k},$$

where, taking s to be the closest vertex to r_i in c_{k-i-k} that also belongs to c_{k-j-k} (we assume w.l.o.g. $s \neq r_j$ since, otherwise, we are in case (I)), p_{s-r_i} is the subpath of c_{k-i-k} that joins r_i with s , p_{k-s} the subpath of c_{k-j-k} that joins s and k without going through r_j , and p_{r_j-k} is the subpath of c_{k-j-k} that joins r_j and k without going through s . By construction, c_{k-e-k} is a cycle around k that includes e and, hence, $e \in N_k$.

Note that the hypothesis on the loop bound is key, since the equivalence class condition does not hold provided it is not fulfilled. Moreover, the converse of the claim is also false. Figure 8 shows a network that serves as a counterexample for both these two statements.

D.2 Characterization of the neighborhoods intersections

Claim 8 (Neighborhoods intersections characterization). *If the loop bound $r_0 \geq 0$ is fulfilled, $i \in \mathcal{V}$, and $j \in \mathcal{V} \cap \left(N_i^{(r_0)} \setminus \{i\}\right)$, then $N_{i \cap j}^{(r_0)} = C_{i-j} \cup P_{C_{i-j}}$, where*

$$\begin{aligned} C_{i-j} &\equiv \{c \subseteq \mathcal{G} \mid c \text{ is a cycle and } i, j \in c\} \\ P_{C_{i-j}} &\equiv \{p \subseteq \mathcal{G} \mid p \text{ is a path and } |\mathcal{V} \cap p \cap c| = 2 \text{ for some } c \in C_{i-j}\}. \end{aligned}$$

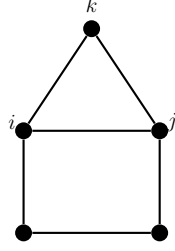


Figure 8: If loop bound is not satisfied, then the equivalence class condition may not hold. In this instance, taking $r_0 = 2$, we have that $k \in N_{i \cap j} = \mathcal{G}$ while $N_k \subsetneq \mathcal{G}$. Moreover, if the equivalence class condition holds for some $r_0 \geq 0$, this does not imply that r_0 is a loop bound. In this instance, although the condition holds for $r_0 = 1$, this is not a fulfilled loop bound.

To prove the claim, we begin by showing that

$$C_{i-j} \cup P_{C_{i-j}} \subseteq N_{i \cap j}^{(r_0)}.$$

By definition, $C_{i-j} \subseteq N_{i \cap j}^{(r_0)}$, while, taking as $i_1, i_2 \in \mathcal{V}$ the intersections of p with some $c \in C_{i-j}$, it is clear that p belongs to a cycle around i (j) that consists of p together with the segment of c that joins i_1 and i_2 through i (j). In particular, $P_{C_{i-j}} \subseteq N_{i \cap j}^{(r_0)}$.

We conclude by showing that $N_{i \cap j}^{(r_0)} \subseteq C_{i-j} \cup P_{C_{i-j}}$. To do so, it suffices to show that

$$\mathcal{E} \cap N_{i \cap j}^{(r_0)} \subseteq \mathcal{E} \cap (C_{i-j} \cup P_{C_{i-j}}).$$

Consider, hence, some $e \in \mathcal{E} \cap N_{i \cap j}^{(r_0)}$ and assume that there is no $c \in C_{i-j}$ such that $e \in c$. By definition, there exist some cycle c_i (c_j) such that $i, e \in c_i$ ($j, e \in c_j$). Moreover, if $i \in c_j$, then $e \in c_j \in C_{i-j}$, a contradiction. Hence, $i \notin c_j$ and we can consider p_{b-i-a} the path contained in c_i that goes through i and has as endpoints a, b the two nodes (one for each branch of c_i) in c_i that are the closest to i and intersect c_j (they must intersect it since $e \in c_i \cap c_j$). Since $a \neq b$ (otherwise $e \notin c_j$, a contradiction), we have that

$$c_{i-j} \equiv p_{b-i-a} \cup p_{a-j-b}$$

is a cycle such that $i, j \in c_{i-j}$, where p_{a-j-b} is the segment of c_j that joins a and b going through j . Since $e \notin c_{i-j}$ by assumption, we have that

$$e \in p_{a-e-b} \equiv c_j \setminus c_{i-j},$$

where p_{a-e-b} consists of the edges in c_j which are not in c_{i-j} together with the endpoints of such edges. This concludes the argument since p_{a-e-b} is a path such that $p_{a-e-b} \cap c_{i-j} = \{a, b\}$.

E Inference in the NIB-method

E.1 r-bounded loops

Given a network $\mathcal{G}$ where the loop bound is fulfilled, the following inference equations are exact:

- For the marginals $p_i(\cdot)$ with $i \in \mathcal{V}$, we have that

$$p_i(x_i) \propto \prod_{i \cap j} m_{\overline{i \cap j} \rightarrow i}(x_i), \quad (17)$$

where we omit a normalization constant that ensures p_i is a probability distribution.

- For the internal energy U , we take the logarithm of the functions in $\mathcal{G}$ as energy terms and we get

$$U = - \sum_{(i,j) \in \mathcal{E}} \frac{1}{Z_{i \cap j}} \text{tr} \left(\log f_{i,j} S_{i \cap j} \prod_{k \in N_{i \cap j}} \prod_{\overline{k \cap q} \neq \overline{i \cap j}} m_{\overline{k \cap q} \rightarrow k} \right),$$

where $Z_{i\cap j}$ is a normalization constant

$$Z_{i\cap j} \equiv \text{tr} \left(S_{i\cap j} \prod_{k \in N_{i\cap j}} \prod_{\overline{k\cap q} \neq \overline{i\cap j}} m_{\overline{k\cap q} \rightarrow k} \right).$$

- For the Shannon entropy S , we have that

$$\begin{aligned} S = & - \sum_{\overline{i\cap j} \in \cap/\sim} \sum_{x_{i\cap j}} p_{i\cap j}(x_{i\cap j}) \log p_{i\cap j}(x_{i\cap j}) \\ & - \sum_{i \in \mathcal{G}} (1 - |\{\overline{k\cap q} \in \cap/\sim \mid i \in \overline{k\cap q}\}|) \sum_{x_i} p_i(x_i) \log p_i(x_i), \end{aligned}$$

where the marginals p_i are computed via (17) and

$$p_{i\cap j}(x_{i\cap j}) \propto S_{i\cap j} \prod_{k \in N_{i\cap j}} \prod_{\overline{k\cap q} \neq \overline{i\cap j}} m_{\overline{k\cap q} \rightarrow k}, \quad (18)$$

where we omit a normalization constant that ensures $p_{i\cap j}$ is a probability distribution.

- For the partition function Z , we have that

$$Z = \frac{\prod_{\overline{i\cap j} \in \cap/\sim} \text{tr} \left(S_{i\cap j} \prod_{k \in N_{i\cap j}} \prod_{\overline{k\cap q} \neq \overline{i\cap j}} m_{\overline{k\cap q} \rightarrow k} \right)}{\prod_{i \in \mathcal{G}} \text{tr} \left(\prod_{\overline{i\cap j}} m_{\overline{i\cap j} \rightarrow i} \right)^{|\{\overline{k\cap q} \in \cap/\sim \mid i \in \overline{k\cap q}\}| - 1}}. \quad (19)$$

For brevity, and since the other equations are at most as difficult as this one, we only show the inference equation for the Shannon entropy S :

To prove the entropy formula, it is sufficient to show that, provided the loop bound is fulfilled, we have that

$$p(x) = \prod_{\overline{i\cap j} \in \cap/\sim} p_{i\cap j}(x_{i\cap j}) \prod_{i \in \mathcal{V}} p_i(x_i)^{1 - |\{\overline{k\cap q} \in \cap/\sim \mid i \in \overline{k\cap q}\}|}. \quad (20)$$

This relation can be proven as follows:

$$\begin{aligned} & \prod_{\overline{i\cap j} \in \cap/\sim} p_{i\cap j}(x_{i\cap j}) \prod_{i \in \mathcal{V}} p_i(x_i)^{1 - |\{\overline{k\cap q} \in \cap/\sim \mid i \in \overline{k\cap q}\}|} \\ &= \prod_{\overline{i\cap j} \in \cap/\sim} \left(\frac{1}{Z_{i\cap j}} S_{i\cap j}(x_{i\cap j}) \prod_{k \in N_{i\cap j}} \prod_{\overline{k\cap q} \neq \overline{i\cap j}} m_{\overline{k\cap q} \rightarrow k}(x_k) \right) \prod_{i \in \mathcal{V}} \left(\frac{1}{Z_i} \prod_{\overline{i\cap j}} m_{\overline{i\cap j} \rightarrow i}(x_i) \right)^{1 - |\{\overline{k\cap q} \in \cap/\sim \mid i \in \overline{k\cap q}\}|} \\ &= \prod_{\overline{i\cap j} \in \cap/\sim} \left(\frac{1}{Z} S_{i\cap j}(x_{i\cap j}) \prod_{k \in N_{i\cap j}} \prod_{\overline{k\cap q} \neq \overline{i\cap j}} n_{\overline{k\cap q} \rightarrow k}(x_k) \right) \prod_{i \in \mathcal{V}} \left(\frac{1}{Z} \prod_{\overline{i\cap j}} n_{\overline{i\cap j} \rightarrow i}(x_i) \right)^{1 - |\{\overline{k\cap q} \in \cap/\sim \mid i \in \overline{k\cap q}\}|} \\ &= \frac{1}{Z} \left(\prod_{\overline{i\cap j} \in \cap/\sim} S_{i\cap j}(x_{i\cap j}) \right) \prod_{i \in \mathcal{V}} \left(\prod_{\overline{i\cap j}} n_{\overline{i\cap j} \rightarrow i}(x_i) \right)^{|\{\overline{k\cap q} \in \cap/\sim \mid i \in \overline{k\cap q}\}| - 1} \prod_{i \in \mathcal{V}} \left(\prod_{\overline{i\cap j}} n_{\overline{i\cap j} \rightarrow i}(x_i) \right)^{1 - |\{\overline{k\cap q} \in \cap/\sim \mid i \in \overline{k\cap q}\}|} \\ &= p(x), \end{aligned}$$

where we have used (18) and (17) in the first equality, together with $Z_{i\cap j}$ and Z_i to refer, respectively, to the missing normalization constants in them. In the second equality, we have used the definition $n_{\overline{k\cap q} \rightarrow k} \equiv m_{\overline{k\cap q} \rightarrow k} / M_{\overline{k\cap q} \rightarrow k}$, with $M_{\overline{k\cap q} \rightarrow k}$ being the product of the normalization constants that have been accumulated in $m_{\overline{k\cap q} \rightarrow k}$ throughout the update process, and we have noted that the product of the missing normalization constants $M_{\overline{k\cap q} \rightarrow k}$ together with $Z_{i\cap j}$ and Z_i , respectively, give us the partition function for each $\overline{i\cap j} \in \cap/\sim$ and $i \in \mathcal{V}$. In the third equality, we have used the hypergraph analog of [22, Theorem 9.1].

E.2 r-unbounded loops

Given a network $\mathcal{G}$ where the loop bound is not fulfilled, we use the following approximate inference equations:

- For the marginals $p_i(\cdot)$ with $i \in \mathcal{V}$, and omitting a normalization constant that ensures p_i is a probability distribution, we use

$$p_i(x_i) \propto \prod_{j \in N_i \setminus \{i\}} \text{tr}_{x_i} \left(S_{i \cap j \setminus \overline{\mathcal{Q}_i}(N_{i \cap j})} \right. \\ \left. \times \prod_{p_1 \in N_{i \cap j} \setminus \{i\}} \prod_{p_2 \in N_{p_1}} m_{p_1 \cap p_2 \rightarrow i \cap j} \right), \quad (21)$$

where $S_{i \cap j \setminus \overline{\mathcal{Q}_i}(N_{i \cap j})}$ stands for the product of the factors which are within $N_{i \cap j}$ and not in $\overline{\mathcal{Q}_i}(N_{i \cap j})$. In order to define $\overline{\mathcal{Q}_i}(\cdot)$, we first consider the set $\mathcal{Q}_i$, which is recursively defined as follows:

- we initialize it as the empty set;
- at each following step, we pick some $j \in N_i \setminus \{i\}$ and we incorporate the functions within $N_{i \cap j}$ to $\mathcal{Q}_i$.

Given this set, we can define a map

$$\overline{\mathcal{Q}_i} : \{N_{i \cap j}\}_{j \in N_i \setminus \{i\}} \rightarrow 2^{\mathcal{E}}$$

such that, for each $j \in N_i \setminus \{i\}$, $\overline{\mathcal{Q}_i}(N_{i \cap j})$ stands for $\mathcal{Q}_i$ at the step right before $j \in \mathcal{V}$ is selected.

- For the internal energy U , we take the logarithm of the factors as energy terms and use

$$U = - \sum_{(i,j) \in \mathcal{E}} \frac{1}{Z_{i \cap j}} \text{tr} \left(\log f_{i,j} S_{i \cap j} \prod_{k \in N_{i \cap j}} \prod_{q \in N_k} m_{k \cap q \rightarrow i \cap j} \right),$$

where $Z_{i \cap j}$ is a normalization constant

$$Z_{i \cap j} = \text{tr} \left(S_{i \cap j} \prod_{k \in N_{i \cap j}} \prod_{q \in N_k} m_{k \cap q \rightarrow i \cap j} \right).$$

- For the Shannon entropy S , we use

$$S = - \sum_{((i,j)) \in \mathcal{G}} \frac{1}{\binom{|S_{i \cap j}|}{2}} \sum_{x_{i \cap j}} p_{i \cap j}(x_{i \cap j}) \log p_{i \cap j}(x_{i \cap j}) \\ - \sum_{(i,j) \in \mathcal{E}} W_{i,j} \sum_{x_i, x_j} p_{i,j}(x_i, x_j) \log p_{i,j}(x_i, x_j) \\ - \sum_{i \in \mathcal{V}} C_i \sum_{x_i} p_i(x_i) \log p_i(x_i), \quad (22)$$

where $W_{i,j}$ and C_i are defined as in (15), the marginals p_i are computed via (21), and, omitting a normalization constant,

$$p_{i \cap j}(x_{i \cap j}) \propto S_{i \cap j}(x_{i \cap j}) \prod_{k \in N_{i \cap j}} \prod_{q \in N_k \setminus N_{i \cap j}} m_{k \cap q \rightarrow i \cap j}(x_k). \quad (23)$$

Moreover, we can compute the two-variable marginals using the intersection neighborhoods, that is, omitting a normalization constant that ensures $p_{i,j}$ is a probability distribution, we have

$$p_{i,j}(x_i, x_j) \propto \text{tr}_{(x_i, x_j)} \left(S_{i \cap j} \prod_{k \in N_{i \cap j}} \prod_{q \in N_k} m_{k \cap q \rightarrow i \cap j} \right).$$

- For the partition function Z , we can use the equation for U and S above together with (13). An alternative approach can also be established by following the discussion in Appendix B.

These approximate inference equations mimic their exact counterparts in Appendix E.1. We simply remark that, in order to obtain the entropy formula (22), we use that

$$\begin{aligned} p(x) &= \prod_{i \cap j \in \cap / \sim} p_{i \cap j}(x_{i \cap j}) \prod_{i \in \mathcal{V}} p_i(x_i)^{1 - |\{ \overline{k \cap q} \in \cap / \sim \mid i \in \overline{k \cap q} \}|} \\ &= \prod_{((i,j)) \in \mathcal{G}} p_{i \cap j}(x_{i \cap j})^{\left(\frac{1}{|S_{i \cap j}|}\right)} \prod_{i \in \mathcal{V}} p_i(x_i)^{1 - \sum_{j \in N_i} \left(\frac{1}{|S_{i \cap j}|}\right)}, \end{aligned}$$

where the first equality is (20) and in the second equality we use the following relations:

$$\begin{aligned} \sum_{j \in N_i \setminus \{i\}} \left(\frac{1}{|S_{i \cap j}| - 1} \right) - 1 &= |\{ \overline{k \cap q} \in \cap / \sim \mid i \in \overline{k \cap q} \}| - 1, \\ \sum_{k, q \in S_{i \cap j}} \left(\frac{1}{|S_{k \cap q}|} \right) &= 1, \end{aligned} \tag{24}$$

which hold provided the loop bound is fulfilled.

F Comparison NIB-method and other message-passing schemes

F.1 Proof of Claim 3

We start by showing sufficiency. To do so, we first note that it is enough to show a speed-up in inference. This is the case since the update complexity until convergence of the NIB-method is always bounded by that of the CN-method. More specifically, this holds for the complexity of each update step since, by hypothesis, and for each $i \in \mathcal{V}$ and $j \in N_i \setminus \{i\}$ for which there exists some (eventually) non-uniform message, there is some $k \in N_i$ such that $N_{i \cap j} \subseteq N_{i \cap k}$. In fact, the role of the (non-trivial) messages $m_{j \rightarrow i}^{(CN)}$ in the CN-method can be decomposed as follows:

$$m_{j \rightarrow i}^{(CN,t)}(x_j) \propto \prod_{\overline{j \cap k} \neq i \cap j} m_{\overline{j \cap k} \rightarrow j}^{(t)}(x_j). \tag{25}$$

Moreover, the number of update steps until convergence in the NIB-method is equal to that in the CN-method. This is the case since the diameter of the NIB-associated hypernetwork is equal to that of each of the members of the family of directed tree networks that we can similarly associate to a network via the CN-method.

We consider now the complexity of inference. To do so, take some $i_0 \in \mathcal{V}$ such that, among all the sets N_i with $i \in \mathcal{V}$, the complexity of summing over N_{i_0} is maximal. We consider two cases: If there is no $j \in \mathcal{V} \cap (N_{i_0} \setminus \{i_0\})$ such that $N_{i_0} = N_{i_0 \cap j}$, then we do not need to sum over N_{i_0} during inference in the NIB-method. Assume, hence, that there is some $j_0 \in \mathcal{V} \cap (N_{i_0} \setminus \{i_0\})$ such that $N_{i_0} = N_{i_0 \cap j_0}$. In this scenario, and provided there exists some $k \in \mathcal{V} \cap (N_{j_0} \setminus N_{i_0 \cap j_0})$, then $N_{i_0} \subsetneq N_{j_0}$ and we contradict the maximality of N_{i_0} . Conversely, if we have that $N_{j_0} \setminus N_{i_0 \cap j_0} = \emptyset$ for any $j_0 \in \mathcal{V}$ such that $N_{i_0} = N_{i_0 \cap j_0}$, then $N_{i_0 \cap j_0} = \mathcal{G}$ by connectedness. This contradicts our hypothesis. Since we can argue along these lines for any $i_0 \in \mathcal{V}$, the NIB-method never sums over the maxima (in terms of time complexity) of $\{N_i\}_{i \in \mathcal{V}}$ during inference and, hence, it has a smaller time complexity than the CN-method.

To show necessity, we can argue by contrapositive. If there is some pair $i, j \in \mathcal{V}$ such that $N_{i \cap j} = \mathcal{G}$, then both the CN and NIB methods have trivial update steps and the complexity of inference is that of summing over the whole network $\mathcal{G}$ for both. Hence, the NIB-method achieves no speed-up.

F.2 Update step time complexity comparison between the CN and NIB methods

Claim 9. *If r_0 is a sharp loop bound, then the r_0 -NIB-method has a smaller message update time complexity than the r_0 -CN-method if and only if, for each pair $i \in \mathcal{V}$, $j \in \mathcal{V} \cap (N_i \setminus \{i\})$ such that, among all the sets $N_{q \cap p}$ with $q \in \mathcal{V}$ and $p \in \mathcal{V} \cap (N_q \setminus \{q\})$, the complexity of summing over $N_{i \cap j}$ is maximal, there is no $k_{i \cap j} \in \mathcal{V}$ such that $N_{i \cap j} = N_{i \cap k_{i \cap j}}$.*

To show sufficiency, we argue by contrapositive, assuming that, for some $N_{i \setminus j}$ with maximal time complexity, there is some $k_{i \setminus j} \in \mathcal{V}$ such that $N_{i \setminus j} = N_{i \cap k_{i \setminus j}}$. This implies that we sum over $N_{i \setminus j}$ in the update steps of the NIB-method and, hence, we have the same time complexity as in the CN-method.

To show necessity, we start by noting that, if we discard the trivial case (where $\mathcal{G} = N_{i \cap j}$ for some $i, j \in \mathcal{V}$ and there are no update steps in either method), then for each pair $i \in \mathcal{V}$, $j \in \mathcal{V} \cap (N_i \setminus \{i\})$, there exists some $q \in \mathcal{V}$ and $p \in \mathcal{V} \cap (N_q \setminus \{q\})$ such that $N_{i \cap j} \subseteq N_{q \setminus p}$. Since, by hypothesis, we have $N_{i \cap j} \subsetneq N_{q \setminus p}$ provided $N_{q \setminus p}$ has maximal time complexity, we get that the time complexity of the update steps in the NIB-method is smaller than that of the CN-method.

F.3 Proof of Claim 4

Before we prove the result, we ought to introduce a definition:

- Given a network $\mathcal{G}$, and a family of subsets of $\mathcal{G}$, $M \equiv \{M_k\}_{k \in \mathcal{K}}$ with $M_k \subseteq \mathcal{G}$ having the property that the endpoints of each edge that belongs to M_k are also in M_k , we can define a BP-type algorithm as follows: We associate a family of messages to each $M_k \in M$:

$$\begin{aligned} & \{m_k^{(M,t)}\}_{k \in \mathcal{K}, t \geq 0}, \\ & m_k : X \rightarrow \mathbb{R}_{\geq 0}. \end{aligned} \tag{26}$$

These messages are uniformly initialized and updated, for each $t \geq 0$, according to

$$m_k^{(M,t+1)}(x_k) \propto \text{tr}_{\setminus x} \left(\prod_{(i,j) \in M_k} f_{i,j} \prod_{k_1 \in I(k)} m_{k_1}^{(M,t)} \right), \tag{27}$$

where x_k is some output variable in $\mathcal{G}$, $I(k) \subseteq \mathcal{K}$, there is at least one message $m_{k_v}^{(M,t)}$ associated to each variable $v \in \mathcal{V} \cap M_k$ that is not the output variable, and every message is used as input in the update of at least one other messages. We call the family of such algorithms the **single variable BP algorithms (SVBP)**. Note that standard BP, the CN-method and the NIB-method are SVBP algorithms. We call a SVBP algorithm **exact** provided it allows us to make exact inference.

Given the definition in the previous paragraph, we can now prove Claim 4. We show the statement by reduction to the absurd.

Assume, in particular, that there is some SVBP algorithm A_0 whose message update time complexity is smaller than that of the NIB-method. Since the update complexity is smaller, there must exist some $i \in \mathcal{V}$ and $j \in N_i \setminus \{i\}$ such that $M_k \subsetneq N_{i \cap j}$ for all $k \in \mathcal{K}$. (This already discards the cases where NIB-method coincides with BP. We assume in what follows that the loop bound r_0 is strictly larger than zero.)

Since A_0 must communicate the information related to i from i to j , there must be some $M_i \in M$ such that $i \in M_i$ and, among the elements in M that include i , M_i is the set used in (26) which is closest (in the sense of the updates (27)) to some $M_j \in M$ such that $j \in M_j$. Because of this, and since the NIB-method is not BP, we have that $M_i \cap (N_{i \cap j} \setminus \{i\}) \neq \emptyset$. Nonetheless, since $M_i \subsetneq N_{i \cap j}$ by hypothesis, there must be some $e \in \mathcal{E} \cap (N_{i \cap j} \setminus M_i)$. In particular, there are at least two nodes $x_1, x_2 \in B(M_i) \cap N_{i \cap j}$, where $B(M_i)$ stands for the **boundary** of M_i , that is, the set of nodes in M_i that are connected to edges outside of M_i .

By definition of the SVBP family, we can split the argument in two cases:

- Both x_1 and x_2 act as input nodes in (26) when updating $m_i^{(M,t)}$. If this is the case, since we can pick x_1 and x_2 such that there is a path between them outside of M_i , there must be, for each of them, a sequence of neighborhoods $S_{x_1}, S_{x_2} \subseteq M$ used in (26) such that, in order for A_0 to be exact, the only intersection between members of different sequences must be the last one in each of them $L_{S_{x_i}} \in S_{x_i}$ for $i = 1, 2$, and the intersection between these two elements must be a single node x_3 . Since x_3 is an input node for both neighborhoods, and to avoid destroying exactness by overcounting, the edges in $L_{S_{x_1}}$ connected to x_3 are not taken into account by $L_{S_{x_2}}$ (and vice versa). This makes the algorithm inexact, and we have reached a contradiction.
- One of them, say x_1 , is an input node when updating $m_i^{(M,t)}$ and the other one is the output node. In this case, we can essentially argue like in the previous case. We should only take into account the fact that, in this scenario, it is possible for x_3 to be an input node of $L_{S_{x_1}}$ and the output node of $L_{S_{x_2}}$. If this is the case, we get a cycle in A_0 and we loose accuracy, reaching again a contradiction.

Since we have just argued that the message update time complexity cannot be smaller than that of the NIB-method, then the inference time complexity must be smaller. We can, however, reach a contradiction along similar lines.

G Inference in the interpolation between the CN and NIB methods

We provide in this section the inference equations for the NIB-DIFF- and CN-NIB-method.

G.1 Inference in the NIB-DIFF-method

For brevity, in this section, we simply state three of the inference equations, and only consider the case where the loop bound is fulfilled. The missing cases can be derived following the ideas in other sections. Moreover, in contrast to the bounded NIB-method, we may have more messages in the NIB-DIFF-method. More specifically, we update non-trivially each message $m_{i \cap j \rightarrow i}^{(t)}$, also those where $N_i \setminus N_{i \cap j} = \emptyset$. This slight change follows from the fact that we intend to use the neighborhoods $N_{i \setminus j}$ during inference and, without the change, some of the inference equations would not be well defined.

Given a network $\mathcal{G}$ where the loop bound is fulfilled, the following inference equations are exact:

- For the marginals $p_i(\cdot)$ with $i \in \mathcal{V}$, we have that

$$p_i(x_i) \propto \text{tr}_{x_i} \left(S_{i \setminus j} \prod_{k \in N_{i \setminus j}} \prod_{\overline{k \cap q} \not\subseteq N_{i \setminus j}} m_{\overline{k \cap q} \rightarrow k} \right),$$

where we omit a normalization constant that ensures p_i is a probability distribution. We assume in this equation that, for each $i \in \mathcal{V}$ and $j \in N_i$ we consider, there exists some $k \in N_i$ such that $\overline{i \cap j} \neq \overline{i \cap k}$. We can do so since, otherwise, the marginal inference is trivial $p_i \equiv m_{\overline{i \cap j} \rightarrow i}$.

- For the internal energy U , we take the logarithm of the functions in $\mathcal{G}$ as energy terms and we get

$$U = - \sum_{(i,j) \in \mathcal{E}} \frac{1}{Z_{i,j}} \text{tr} \left(\log f_{i,j} S_{a_{i,j} \setminus b_{i,j}} \right. \\ \left. \times \prod_{p_1 \in N_{a_{i,j} \setminus b_{i,j}}} \prod_{\overline{p_1 \cap p_2} \not\subseteq N_{a_{i,j} \setminus b_{i,j}}} m_{\overline{p_1 \cap p_2} \rightarrow p_1} \right)$$

where we pick $a_{i,j}, b_{i,j} \in \mathcal{V}$ such that $(i,j) \in S_{a_{i,j} \setminus b_{i,j}}$ ¹⁴, and $Z_{i,j}$ is a normalization constant

$$Z_{i,j} \equiv \text{tr} \left(S_{a_{i,j} \setminus b_{i,j}} \prod_{p_1 \in N_{a_{i,j} \setminus b_{i,j}}} \prod_{\overline{p_1 \cap p_2} \not\subseteq N_{a_{i,j} \setminus b_{i,j}}} m_{\overline{p_1 \cap p_2} \rightarrow p_1} \right).$$

- For the partition function Z , we have that

$$Z = \frac{\prod_{i \in \mathcal{V}} \prod_{j \in N_i \setminus \{i\}} \text{tr} \left(S_{i \setminus j} \prod_{k \in N_{i \setminus j}} \prod_{\overline{k \cap q} \not\subseteq N_{i \setminus j}} m_{\overline{k \cap q} \rightarrow k} \right)}{\prod_{i \cap j \in \mathcal{V}} \text{tr} \left(S_{i \cap j} \prod_{k \in N_{i \cap j}} \prod_{\overline{k \cap q} \neq \overline{i \cap j}} m_{\overline{k \cap q} \rightarrow k} \right)^{1 - \sum_{k \in i \cap j} |\{ \overline{k \cap q} | \overline{k \cap q} \neq \overline{i \cap j} \}|}}.$$

G.2 Inference in the CN-NIB-method

As in the previous section, we simply state three of the inference equations, and only consider the case where the loop bound is fulfilled. In contrast to the update equations in the CN-method, we include here more messages. More specifically, for each pair $i \in \mathcal{V}$, $j \in N_i$ such that $N_{i \setminus j} = \emptyset$, we include a message from i to some **virtual** node $v_{i,j}$, for which we only need to assume $N_{i \setminus v_{i,j}} = N_{i \cap j}$, and which we define

¹⁴This only excludes the case where $\mathcal{G} = N_{i \cap j}$ for $i, j \in \mathcal{V}$, where the NIB-DIFF-method would essentially coincide with the NIB-method.

and update in the CN fashion as if $v_{i,j}$ was an actual node. These messages are only relevant for the computation of marginals and are ignored in the other inference equations.

Given a network $\mathcal{G}$ where the loop bound is fulfilled, the following inference equations are exact:

- For the marginals $p_i(\cdot)$ with $i \in \mathcal{V}$, omitting a normalization constant that ensures p_i is a probability distribution, we have that

$$p_i(x_i) \propto \prod_{j \in N_i \setminus \{i\}} \left(m_{i \rightarrow j}^{(\text{CN})}(x_i) \right)^{\left((|S_{i \cap j}|-1) \left(\sum_{j \in N_i} \left(\frac{1}{|S_{i \cap j}|-1} \right)^{-1} \right) \right)^{-1}}.$$

- For the internal energy U , we take the logarithm of the functions in $\mathcal{G}$ as energy terms and we get

$$U = - \sum_{(i,j) \in \mathcal{E}} \frac{1}{Z_{i \cap j}} \text{tr} \left(\log f_{i,j} S_{i \cap j} m_{i \rightarrow j}^{(\text{CN})} \prod_{k \in N_{i \cap j} \setminus \{i\}} m_{k \rightarrow i}^{(\text{CN})} \right),$$

where $Z_{i \cap j}$ is a normalization constant

$$Z_{i \cap j} \equiv \text{tr} \left(S_{i \cap j} m_{i \rightarrow j}^{(\text{CN})} \prod_{k \in N_{i \cap j} \setminus \{i\}} m_{k \rightarrow i}^{(\text{CN})} \right).$$

- For the partition function Z , we have that

$$Z = \frac{\prod_{((i,j)) \in \mathcal{G}} \text{tr} \left(S_{i \cap j} m_{j \rightarrow i}^{(\text{CN})} \prod_{k \in N_{i \cap j} \setminus \{j\}} m_{k \rightarrow j}^{(\text{CN})} \right)^{\left(\frac{1}{|S_{i \cap j}|} \right)}}{\prod_{i \in \mathcal{V}_{\text{NT}}} \text{tr} \left(\prod_{j \in N_i \setminus \{i\}} \left(m_{i \rightarrow j}^{(\text{CN})} \right)^{\left((|S_{i \cap j}|-1) \left(\sum_{j \in N_i \setminus \{i\}} \left(\frac{1}{|S_{i \cap j}|-1} \right)^{-1} \right) \right)^{-1}} \right)^{\sum_{j \in N_i \setminus \{i\}} \left(\frac{1}{|S_{i \cap j}|-1} \right)^{-1}}},$$

where

$$\mathcal{V}_{\text{NT}} \equiv \{i \in \mathcal{V} : |N_i \setminus \{i\}| \geq 2\}.$$

For brevity, and since the other equations are at most as difficult as this one, we only show the

inference equation for the partition function Z , which can be derived via the following chain of equalities:

$$\begin{aligned}
& \frac{\prod_{((i,j)) \in \mathcal{G}} \text{tr} \left(S_{i \cap j} m_{j \rightarrow i}^{(\text{CN})} \prod_{k \in N_{i \cap j} \setminus \{j\}} m_{k \rightarrow j}^{(\text{CN})} \right)^{\frac{1}{2^{|S_{i \cap j}|}}}}{\prod_{i \in \mathcal{V}_{\text{NT}}} \text{tr} \left(\prod_{j \in N_i \setminus \{i\}} \left(m_{i \rightarrow j}^{(\text{CN})} \right)^{\left((|S_{i \cap j}| - 1) \left(\sum_{j \in N_i \setminus \{i\}} \left(\frac{1}{|S_{i \cap j}| - 1} \right) - 1 \right) \right)^{-1}} \right)^{\sum_{j \in N_i \setminus \{i\}} \left(\frac{1}{|S_{i \cap j}| - 1} \right) - 1}} \\
&= \frac{\prod_{\overline{i \cap j}} \text{tr} \left(S_{i \cap j} m_{j \rightarrow i}^{(\text{CN})} \prod_{k \in N_{i \cap j} \setminus \{j\}} m_{k \rightarrow j}^{(\text{CN})} \right)}{\prod_{i \in \mathcal{V}_{\text{NT}}} \text{tr} \left(\prod_{j \in (N_i \setminus \{i\}) / \sim} \left(m_{i \rightarrow j}^{(\text{CN})} \right)^{\left(\sum_{j \in N_i \setminus \{i\}} \left(\frac{1}{|S_{i \cap j}| - 1} \right) - 1 \right)^{-1}} \right)^{\sum_{j \in N_i \setminus \{i\}} \left(\frac{1}{|S_{i \cap j}| - 1} \right) - 1}} \\
&= \frac{\prod_{\overline{i \cap j}} \text{tr} \left(S_{i \cap j} m_{j \rightarrow i}^{(\text{CN})} \prod_{k \in N_{i \cap j} \setminus \{j\}} m_{k \rightarrow j}^{(\text{CN})} \right)}{\prod_{i \in \mathcal{V}_{\text{NT}}} \text{tr} \left(\prod_{j \in (N_i \setminus \{i\}) / \sim} \left(m_{i \rightarrow j}^{(\text{CN})} \right)^{\left((|j \in N_i \setminus \{i\}| / \sim - 1)^{-1} \right)^{|j \in N_i \setminus \{i\}| / \sim - 1}} \right)} \tag{28} \\
&= \frac{\prod_{\overline{i \cap j}} \text{tr} \left(S_{i \cap j} n_{j \rightarrow i}^{(\text{CN})} \prod_{k \in N_{i \cap j} \setminus \{j\}} n_{k \rightarrow j}^{(\text{CN})} \right)}{\prod_{i \in \mathcal{V}_{\text{NT}}} \text{tr} \left(\prod_{j \in (N_i \setminus \{i\}) / \sim} \left(n_{i \rightarrow j}^{(\text{CN})} \right)^{\left((|j \in N_i \setminus \{i\}| / \sim - 1)^{-1} \right)^{|j \in N_i \setminus \{i\}| / \sim - 1}} \right)} \\
&= \frac{\prod_{\overline{i \cap j}} Z}{\prod_{i \in \mathcal{V}_{\text{NT}}} \text{tr} \left(\prod_{j \in (N_i \setminus \{i\}) / \sim} \left(\prod_{k \in (N_i \setminus N_j) / \sim} n_{i \cap k \rightarrow i} \right)^{\left((|j \in N_i \setminus \{i\}| / \sim - 1)^{-1} \right)^{|j \in N_i \setminus \{i\}| / \sim - 1}} \right)} \\
&= \frac{\prod_{\overline{i \cap j}} Z}{\prod_{i \in \mathcal{V}_{\text{NT}}} \text{tr} \left(\prod_{k \in (N_i \setminus \{i\}) / \sim} n_{i \cap k \rightarrow i} \right)^{|j \in N_i \setminus \{i\}| / \sim - 1}} \\
&= \frac{\prod_{\overline{i \cap j}} Z}{\prod_{i \in \mathcal{V}_{\text{NT}}} Z^{|j \in N_i \setminus \{i\}| / \sim - 1}} \\
&= Z,
\end{aligned}$$

where, in the first equality, we use the second equality in (24) together with the fact that, for all $i, j, k \in \mathcal{G}$, $m_{i \rightarrow j}^{(\text{CN})} = m_{i \rightarrow k}^{(\text{CN})}$ provided $S_{i \cap j} = S_{i \cap k}$. In the second equality, we use the first equality in (24). In the third equality we use the notation $n_{i \rightarrow j}^{(\text{CN})}$ to refer to $m_{i \rightarrow j}^{(\text{CN})}$ without the normalization constants and note that, given the powers in the denominator, each normalization constant $M_{i \rightarrow j}^{(\text{CN})}$ in the upper part gets cancelled with one in the lower part. In the fourth equality, we decompose the messages $n_{i \rightarrow j}^{(\text{CN})}$ into their constituent intersection messages (that is, into the product of the unnormalized messages in the NIB-method that carry the same information as $n_{i \rightarrow j}^{(\text{CN})}$). In the last equality, we use a result analogous to [22, Theorem 9.1] for hypergraphs.