Abstract

We analyze variational inference for highly symmetric graphical models such as those arising from first-order probabilistic models. We first show that for these graphical models, the tree-reweighted variational objective lends itself to a compact lifted formulation which can be solved much more efficiently than the standard TRW formulation for the ground graphical model. Compared to earlier work on lifted belief propagation, our formulation leads to a convex optimization problem for lifted marginal inference and provides an upper bound on the partition function. We provide two approaches for improving the lifted TRW upper bound. The first is a method for efficiently computing maximum spanning trees in highly symmetric graphs, which can be used to optimize the TRW edge appearance probabilities. The second is a method for tightening the relaxation of the marginal polytope using lifted cycle inequalities and novel exchangeable cluster consistency constraints.

1 Introduction

Lifted probabilistic inference focuses on exploiting symmetries in probabilistic models for efficient inference [5, 2, 3, 10, 17, 18, 21]. Work in this area has demonstrated the possibility to perform very efficient inference in highly-connected, large tree-width, but symmetric models, such as those arising in the context of relational (first-order) probabilistic models and exponential family random graphs [19]. These models also arise frequently in probabilistic programming languages, an area of increasing importance as demonstrated by DARPA’s PPAML program (Probabilistic Programming for Advancing Machine Learning).

Even though lifted inference can sometimes offer order-of-magnitude improvement in performance, approximation is still necessary. A topic of particular interest is the interplay between lifted inference and variational approximate inference. Lifted loopy belief propagation (LBP) [13, 21] was one of the first attempts at exploiting symmetry to speed up loopy belief propagation; subsequently, counting belief propagation (CBP) [16] provided additional insights into the nature of symmetry in BP. Nevertheless, these work were largely procedural and specific to the choice of message-passing algorithm (in this case, loopy BP). More recently, Bui et al., [3] proposed a general framework for lifting a broad class of convex variational techniques by formalizing the notion of symmetry (defined via automorphism groups) of graphical models and the corresponding variational optimization problems themselves, independent of any specific methods or solvers.

Our goal in this paper is to extend the lifted variational framework in [3] to address the important case of approximate marginal inference. In particular, we show how to lift the tree-reweighted (TRW) convex formulation of marginal inference [28]. As far as we know, our work presents the first lifted convex variational marginal inference, with the following benefits over previous work: (1) a lifted convex upper bound of the log-partition function, (2) a new tightening of the relaxation of the lifted marginal polytope exploiting exchangeability, and (3) a convergent inference algorithm. We note that convex upper bounds of the log-partition function immediately lead to concave lower bounds of the log-likelihood which can serve as useful surrogate loss functions in learning and parameter estimation [29, 13].

To achieve the above goal, we first analyze the symmetry of the TRW log-partition and entropy bounds. Since TRW bounds depend on the choice of the edge appearance probabilities $\rho$, we prove that the quality of the TRW bound is not affected if one only works with suitably symmetric $\rho$. Working with symmetric $\rho$ gives rise to an explicit lifted formulation of the TRW optimization problem that is equivalent but much more compact. This convex objective function can be convergently optimized via a Frank-Wolfe (conditional gradient) method where each Frank-Wolfe iteration solves a lifted MAP inference problem. We then
discuss the optimization of the edge-appearance vector $\rho$, effectively yielding a lifted algorithm for computing maximum spanning trees in symmetric graphs.

As in Bui et al.’s framework, our work can benefit from any tightening of the local polytope such as the use of cycle inequalities [1, 23]. In fact, each method for relaxing the marginal polytope immediately yields a variant of our algorithm. Notably, in the case of exchangeable random variables, radically sharper tightening (sometimes even exact characterization of the lifted marginal polytope) can be obtained via a set of simple and elegant linear constraints which we call exchangeable polytope constraints. We provide extensive simulation studies comparing the behaviors of different variants of our algorithm with exact inference (when available) and lifted LBP demonstrating the advantages of our approach. The supplementary material [4] provides additional proof and algorithm details.

2 Background

We begin by reviewing variational inference and the tree-reweighted (TRW) approximation. We focus on inference in Markov random fields, which are distributions in the exponential family given by $P_t(x; \theta) = \exp \{ \langle \Phi(x), \theta \rangle - A(\theta) \}$, where $A(\theta)$ is called the log-partition function and serves to normalize the distribution.

We assume that the random variables $x \in \mathcal{X}^n$ are discrete-valued, and that the features $(\Phi_i, i \in \mathcal{I})$ factor according to the graphical model structure $G$. $\Phi$ can be non-pairwise and is assumed to be overcomplete. This paper focuses on the inference tasks of estimating the marginal probabilities $p(x_i)$ and approximating the log-partition function. Throughout the paper, the domain $\mathcal{X}$ is the binary domain $\{0, 1\}$, however, except for the construction of exchangeable polytope constraints in Section 6, this restriction is not essential.

Variational inference approaches view the log-partition function as a convex optimization problem over the marginal polytope $A(\theta) = \sup_{\rho \in \mathcal{M}(G)} \{ \langle \mu, \theta \rangle - A^*(\mu) \}$ and seek tractable approximations of the negative entropy $A^*$ and the marginal polytope $\mathcal{M}$ [27]. Formally, $- A^*(\mu)$ is the entropy of the maximum entropy distribution with moments $\mu$. Observe that $- A^*(\mu)$ is upper bounded by the entropy of the maximum entropy distribution consistent with any subset of the expected sufficient statistics $\mu$. To arrive at the TRW approximation [26], one uses a subset given by the pairwise moments of a spanning tree\(^1\). Hence for any distribution $\rho$ over spanning trees, an upper bound on $- A^*$ is obtained by taking a convex combination of tree entropies $- B^*(\tau, \rho) = \sum_{e \in V(G)} H(\tau_e) - \sum_{e \in E(G)} I(\tau_e) \rho_e$. Since $\rho$ is a distribution over spanning trees, it must belong to the spanning tree polytope $\mathcal{P}(G)$ with $\rho_e$ denoting the edge appearance probability of $e$. Combined with a relaxation of the marginal polytope $\mathcal{M}$, an upper bound $B$ of the log-partition function is obtained:

$$A(\theta) \leq B(\theta, \rho) = \sup_{\tau \in \mathcal{P}(G)} \langle \tau, \theta \rangle - B^*(\tau, \rho) \tag{1}$$

We note that $B^*$ is linear w.r.t. $\rho$, and for $\rho \in \mathcal{P}(G)$, $B^*$ is convex w.r.t. $\tau$. On the other hand, $B$ is convex w.r.t. $\rho$ and $\theta$.

The optimal solution $\tau^*(\rho, \theta)$ of the optimization problem (1) can be used as an approximation to the mean parameter $\mu(\theta)$. Typically, the local polytope $\mathcal{L}(\theta)$ given by pairwise consistency constraints is used as the relaxation $\mathcal{M}$; in this paper, we also consider tightening of the local polytope.

Since (1) holds with any edge appearance $\rho$ in the spanning tree polytope $\mathcal{T}$, the TRW bound can be further improved by optimizing $\rho$:

$$\inf_{\rho \in \mathcal{T}(G)} B(\theta, \rho) \tag{2}$$

The resulting $\rho^*$ is then plugged into (1) to find the marginal approximation. In practice, one might choose to work with some fixed choice of $\rho$, for example the uniform distribution over all spanning trees. [14] proposed using the most uniform edge-weight $\arg \inf_{\rho \in \mathcal{T}(G)} \sum_{e \in E} (\rho_e - \frac{1}{|E|})^2$ which can be found via conditional gradient where each direction-finding step solves a maximum spanning tree problem.

Several algorithms have been proposed for optimizing the TRW objective (1) given fixed edge appearance probabilities. [27] derived the tree-reweighted belief propagation algorithm from the fixed point conditions. [8] show how to solve the dual of the TRW objective, which is a geometric program. Although this algorithm has the advantage of guaranteed convergence, it is non-trivial to generalize this approach to use tighter relaxations of the marginal polytope, which we show is essential for lifted inference. [14] use an explicit set of spanning trees and then use dual decomposition to solve the optimization problem. However, as we show in the next section, to maintain symmetry it is essential that one not work directly with spanning trees but rather use symmetric edge appearance probabilities. [23] optimize TRW over the local and cycle polytopes using a Frank-Wolfe (conditional gradient) method, where each iteration requires solving a linear program. We follow this latter approach in our paper.

To optimize the edge appearance in (2), [26] proposed using conditional gradient. They observed that $\frac{\partial B(\theta, \rho)}{\partial \rho_e} = - \frac{\partial B^*(\tau^*_e, \rho)}{\partial \rho_e} = - I(\tau^*_e)$ where $\tau^*$ is the solution of (1). The direction-finding step in conditional gradient reduces to solving $\sup_{\rho \in \mathcal{T}} \langle \rho, I \rangle$, again equivalent to finding the maximum spanning tree with edge mutual information $I(\tau^*_e)$ as weights. We discuss the corresponding lifted problem in section 5.

\(^1\)If the original model contains non-pairwise potentials, they can be represented as cliques in the graphical model, and the bound based on spanning trees still holds.
3 Lifted Variational Framework

We build on the key element of the lifted variational framework introduced in [3]. The automorphism group of a graphical model, or more generally, an exponential family is defined as the group $\mathbb{A}$ of permutation pairs $(\pi, \gamma)$ where $\pi$ permutes the set of variables and $\gamma$ permutes the set of features in such a way that they preserve the feature function: $\Phi^{-1}(x^\pi) = \Phi(x)$. Note that this construction of $\mathbb{A}$ is entirely based on the structure of the model and does not depend on the particular choice of the model parameters; nevertheless the group stabilizes the key characteristics of the exponential family such as the marginal polytope $M$, the log-partition $A$ and entropy $-A^*$. As shown in [3] the automorphism group is particularly useful for exploiting symmetries when parameters are tied. For a given parameter-tying partition $\Delta$ such that $\theta_i = \theta_j$ for $i, j$ in the same cell of $\Delta$, the group $\mathbb{A}$ gives rise to a subgroup called the lifting group $\mathbb{A}_\Delta$ that stabilizes the tied-parameter vector $\theta$ as well as the exponential family. The orbit partition of the the lifting group can be used to formulate equivalent but more compact variational problems. More specifically, let $\varphi = \varphi(\Delta)$ be the orbit partition induced by the lifting group on the feature index set $I = \{1 \ldots m\}$, let $\mathbb{R}^m_\varphi$ denote the symmetrized subspace $\{r \in \mathbb{R}^m \text{ s.t. } r_i = r_j \forall i, j \text{ in the same cell of } \varphi\}$ and define the lifted marginal polytope $M(\varphi)$ as $M \cap \mathbb{R}^m_\varphi$, then (see Theorem 4 of [3])

$$\sup_{\mu \in \mathcal{M}} \langle \theta, \mu \rangle - A^*(\mu) = \sup_{\mu \in M(\varphi)} \langle \theta, \mu \rangle - A^*(\mu) \quad (3)$$

In practice, we need to work with convex variational approximations of the LHS of (3) where $\mathcal{M}$ is relaxed to an outer bound $\text{OUTER}(\mathcal{G})$ and $A^*$ is approximated by a convex function $B^*(\mu)$. We now state a similar result for lifting general convex approximations.

**Theorem 1.** If $B^*(\mu)$ is convex and stabilized by the lifting group $\mathbb{A}_\Delta$, i.e., for all $(\pi, \gamma) \in \mathbb{A}_\Delta$, $B^*(\mu^\pi) = B^*(\mu)$, then $\varphi$ is the lifting partition for the approximate variational problem

$$\sup_{\mu \in \text{OUTER}(\mathcal{G})} \langle \theta, \mu \rangle - B^*(\mu) = \sup_{\mu \in \text{OUTER}(\varphi)} \langle \theta, \mu \rangle - B^*(\mu) \quad (4)$$

The importance of Theorem 1 is that it shows that it is equivalent to optimize over a subset of $\text{OUTER}(\mathcal{G})$ where pseudo-marginals in the same orbit are restricted to take the same value. As we will show in Section 4.2, this will allow us to combine many of the terms in the objective, which is where the computational gains will derive from.

\[ \text{FORMALLY, } \mathbb{G} \text{ stabilizes } x \text{ if } x^g = x \text{ for all } g \in \mathbb{G}. \]

\[ \text{IF } \Delta = \{\Delta_1, \ldots, \Delta_K\} \text{ is a partition of } S, \text{ then each subset } \Delta_k \subset S \text{ is called a cell.} \]

4 Lifted Tree Reweighted Problem

4.1 Symmetry of TRW Bounds

We now show that Theorem 1 can be used to lift the TRW optimization problem (1). Note that the applicability of Theorem 1 is not immediately obvious since $B^*$ depends on the distribution over trees implicit in $\rho$. In establishing that the condition in Theorem 1 holds, we need to be careful so that the choice of the distribution over trees $\rho$ does not destroy the symmetry of the problem.

The result below ensures that with no loss in optimality, $\rho$ can be assumed to be suitably symmetric. More specifically, let $\varphi^\rho = \varphi^\rho(\Delta)$ be the set of $\mathcal{G}$’s edge orbits induced by the action of the lifting group $\mathbb{A}_\Delta$; the edge-weights $\rho_e$ for every $e$ in the same edge orbits can be constrained to be the same, i.e. $\rho$ can be restricted to $T[\varphi^\rho]$.

**Theorem 2.** For any $\rho \in T$, there exists a symmetrized $\hat{\rho} \in T[\varphi^\rho]$ that yields at least as good an upper bound, i.e.

$$B(\theta, \hat{\rho}) \leq B(\theta, \rho) \forall \theta \in \Theta|\Delta|$$

As a consequence, in optimizing the edge appearance, $\rho$ can be restricted to the symmetrized spanning tree polytope $T[\varphi^\rho]$.

$$\forall \theta \in \Theta|\Delta|, \inf_{\rho \in T} B(\theta, \rho) = \inf_{\rho \in T[\varphi^\rho]} B(\theta, \rho)$$

**Proof.** Let $\rho$ be the argmin of the LHS, and define $\hat{\rho} = \frac{1}{|\mathbb{A}_\Delta|} \sum_{\pi \in \mathbb{A}_\Delta} \rho^\pi$ so that $\hat{\rho} \in T[\varphi^\rho]$. For all $(\pi, \gamma) \in \mathbb{A}_\Delta$ and for all tied-parameter $\theta \in \Theta|\Delta|$, $\theta^\pi = \theta$, so $B(\theta, \rho^\pi) = B(\theta^\pi, \rho^\pi)$. By theorem 1 of [3], $\pi$ must be an automorphism of the graph $\mathcal{G}$. By lemma 7 (see supplementary material), $B(\theta^\pi, \rho^\pi) = B(\theta, \rho)$. Thus $B(\theta, \rho^\pi) = B(\theta, \rho)$. Since $B$ is convex w.r.t. $\rho$, by Jensen’s inequality we have that $B(\theta, \hat{\rho}) \leq \frac{1}{|\mathbb{A}_\Delta|} \sum_{\pi \in \mathbb{A}_\Delta} B(\theta, \rho^\pi) = B(\theta, \rho)$. \( \square \)

Using a symmetric choice of $\rho$, the TRW bound $B^*$ then satisfies the condition of theorem 1, enabling the applicability of the general lifted variational inference framework.

**Theorem 3.** For a fixed $\tau \in T[\varphi^\rho]$, $\varphi$ is the lifting partition for the TRW variational problem

$$\sup_{\tau \in \text{OUTER}(\mathcal{G})} \langle \tau, \theta \rangle - B^*(\tau, \rho) = \sup_{\tau \in \text{OUTER}(\varphi)} \langle \tau, \theta \rangle - B^*(\tau, \rho) \quad (5)$$
4.2 Lifted TRW Problems

We give the explicit lifted formulation of the TRW optimization problem (5). As in [3], we restrict $\tau$ to $\text{OUTER}_\varphi$ by introducing the lifted variables $\tilde{\tau}_j$ for each cell $\varphi_j$, and for all $i \in \varphi_j$, enforcing that $\tilde{\tau}_i = \varphi_j$. Effectively, we substitute every occurrence of $\tau_i, i \in \varphi_j$ by $\tilde{\tau}_j$; in vector form, $\tau$ is substituted by $D\tilde{\tau}$ where $D$ is the characteristic matrix of the partition $\varphi$; $D_{ij} = 1$ if $i \in \varphi_j$ and 0 otherwise. This results in the lifted form of the TRW problem

$$\sup_{D\tilde{\tau} \in \text{OUTER}} \langle \tilde{\tau}, \tilde{\theta} \rangle - B^*(\tilde{\tau}, \tilde{\rho})$$ (6)

where $\tilde{\theta} = D^T\theta$; $B^*$ is obtained from $B^*$ via the above substitution; and $\tilde{\rho}$ is the edge appearance per edge-orbit: for every degree orbit $e$, and for every edge $e \in e$, $\rho_e = \tilde{\rho}_e$. Using an alternative but equivalent form $B^* = -\sum_{v \in V} (1 - \sum_{e \in \mathcal{E}(v)} \rho_{e}) H(\tau_{v}) - \sum_{e \in \mathcal{E}} e(H(\tau_{e}))$, we obtain the following explicit form for

$$B^*(\tilde{\tau}, \tilde{\rho}) \begin{aligned} &= -\sum_{v \in V} \left( |v| - \sum_{e \in \mathcal{E}(v)} |e| d(v, e) \tilde{\rho}_e \right) H(\tilde{\tau}_v) - \sum_{e \in \mathcal{E}} |e| \tilde{\rho}_e H(\tilde{\tau}_e) \end{aligned}$$ (7)

Intuitively, the above can be viewed as a combination of node and edge entropies defined on nodes and edges of the lifted graph $\tilde{G}$. Nodes of $\tilde{G}$ are the node orbits of $G$ while edges are the edge-orbits of $\tilde{G}$. $\tilde{G}$ is not a simple graph: it can have self-loops or multi-edges between the same node pair (see Fig. 1). We encode the incidence on this graph as follows: $d(v, e) = 0$ if $v$ is not incident to $e$, $d(v, e) = 1$ if $v$ is incident to $e$, $v$ and $e$ is not a loop, $d(v, e) = 2$ if $e$ is a loop incident to $v$. The entropy at the node orbit $v$ is defined as

$$H(\tilde{\tau}_v) = -\sum_{t \in X} \tilde{\tau}_{v,t} \ln(\tilde{\tau}_{v,t})$$

and the entropy at the edge orbit $e$ is

$$H(\tilde{\tau}_e) = -\sum_{t,h \in X} \tilde{\tau}_{e,t,h} \ln(\tilde{\tau}_{e,t,h})$$

where $\{e_1, e_2\}$ for $e_1, e_2 \in V$ is a representative (any element) of $e$, $\{e_1:t, e_2:h\}$ is the assignment of the ground edge $\{e_1, e_2\}$, and $\{e_1:t, e_2:h\}$ is the assignment orbit. As in [3], we write $\{e_1:t, e_2:t\}$ as $e:t$, and for $t < h$, $\{e_1:t, e_2:h\}$ as $a:(t,h)$ where $a$ is the arc-orbit $\{e_1, e_2\}$. When $\text{OUTER}$ is the local or cycle polytope, the constraints $D\tilde{\tau} \in \text{OUTER}$ yield the lifted local (or cycle) polytope respectively. For these constraints, we use the same form given in [3]. In section 6, we describe a set of additional constraints for further tightening when some cluster of nodes are exchangeable.

**Example.** Consider the MRF shown in Fig. 1 (left) with 10 binary variables that we denote $B_i$ (for the blue nodes) and $R_i$ (for the red nodes). The node and edge coloring denotes shared parameters. Let $\theta_b$ and $\theta_r$ be the single-node potentials used for the blue and red nodes, respectively. Let $\theta_e$ be the edge potential used for the red edges connecting the blue and red nodes, $\theta_{b_e}$ for the edge potential used for the blue edges $(B_i, B_{i+1})$, and $\theta_{r_e}$ for the edge potential used for the black edges $(B_i, B_{i+2})$.

There are two node orbits: $b = \{B_1, \ldots, B_5\}$ and $r = \{R_1, \ldots, R_5\}$. There are three edge orbits: $e_b$ for the red edges, $e_b$ for the blue edges, and $e_k$ for the black edges. The size of the node and edge orbits are all 5 (e.g., $|b| = |b| = 5$), and $d(b, e_b) = d(r, e_r) = 1$, $d(b, e_b) = d(b, e_k) = 2$. Suppose that $\rho$ corresponds to a uniform distribution over spanning trees, which satisfies the symmetry needed by Theorem 2. We then have $\tilde{\rho}_e = 1$ and $\tilde{\rho}_b = \tilde{\rho}_k = 2$. Putting all of this together, the lifted TRW entropy is given by $B^*(\tilde{\tau}, \tilde{\rho}) = 8H(\tilde{\tau}_{b}) - 5H(\tau_{r_e}) - 2H(\tau_{e_b}) - 2H(\tau_{e_k})$. We illustrate the expansion of the entropy of the red edge orbit $H(\tilde{\tau}_{e_b})$. This edge orbit has 2 corresponding arc-orbits: $r_{b_a} = \{(R_i, B_i)\}$ and $r_{b_a} = \{(B_i, R_i)\}$. Thus, $H(\tilde{\tau}_{e_b}) = -\tilde{\tau}_{e_b:00} \ln \tilde{\tau}_{e_b:00} - \tilde{\tau}_{e_b:11} \ln \tilde{\tau}_{e_b:11} - \tilde{\tau}_{r_{b_a}:00} \ln \tilde{\tau}_{r_{b_a}:00} - \tilde{\tau}_{r_{b_a}:11} \ln \tilde{\tau}_{r_{b_a}:11}$

Finally, the linear term in the objective is given by $\langle \tilde{\tau}, \tilde{\theta} \rangle = 5(\tilde{\tau}_{b, \theta_b}) + 5(\tilde{\tau}_{r, \theta_r}) + 5(\tilde{\tau}_{e_b, \theta_{e_b}}) + 5(\tilde{\tau}_{e_k, \theta_{e_k}})$.

4.3 Optimization using Frank-Wolfe

What remains is to describe how to optimize Eq. 6. Our lifted tree-reweighted algorithm is based on Frank-Wolfe, also known as theconditional gradient method [7, 11]. First, we initialize with a pseudo-marginal vector corresponding to the uniform distribution, which is guaranteed to be in the lifted marginal polytope. Next, we solve the linear program whose objective is given by the gradient of the objective Eq. 6 evaluated at the current point, and whose constraint set is $\text{OUTER}$. When using the lifted cycle relaxation, we solve this linear program using a cutting-plane algorithm [3, 23]. We then perform a line search to find the optimal step size using a golden section search (a type of binary search that finds the maxima of a unimodal function), and finally repeat using the new pseudo-marginal vector.

We warm start each linear program using the optimal basis found in the previous run, which makes the LP solves ex-
5 Lifted Maximum Spanning Tree

Optimizing the TRW edge appearance probability \( \rho \) requires finding the maximum spanning tree (MST) in the ground graphical model. For lifted TRW, we need to perform MST while using only information from the node and edge orbits, without referring to the ground graph. In this section, we present a lifted MST algorithm for symmetric graphs which works at the orbit level.

Suppose that we are given a weighted graph \((G, w)\), its automorphism group \( \mathbb{A} = Aut(G) \) and its node and edge orbits. We aim to derive an algorithm analogous to the Kruskal’s algorithm, but with complexity depends only on the number of node/edge orbits of \( G \). However, if the algorithm has to return an actual spanning tree of \( G \) then clearly its complexity cannot be less than \( O(|V|) \). Instead, we consider an equivalent problem: solving a linear program on the spanning-tree polytope

\[
\sup_{\rho \in \bar{T}(G)} \langle \rho, w \rangle \tag{8}
\]

The same mechanism for lifting convex optimization problem (Lemma 1 in [3]) applies to this problem. Let \( \mathcal{E}_k \) be the edge orbit partition, then an equivalent lifted problem problem is

\[
\sup_{\rho \in \bar{T}(\rho, w)} \langle \rho, w \rangle \tag{9}
\]

Since \( \rho_c \) is constrained to be the same for edges in the same orbit, it is now possible to solve (9) with complexity depending only on the number of orbits. Any solution \( \rho \) of the LP (8) can be turned into a solution \( \bar{\rho} \) of (9) by letting \( \bar{\rho}(e) = \frac{1}{|e|} \sum_{e' \in \rho_e} \rho(e') \).

5.1 Lifted Kruskal’s Algorithm

The Kruskal’s algorithm first sorts the edges according to their decreasing weight. Then starting from an empty graph, at each step it greedily attempts to add the next edge while maintaining the property that the used edges form a forest (containing no cycle). The forest obtained at the end of this algorithm is a maximum weight spanning tree.

Imagine how Kruskal’s algorithm would operate on a weighted graph \( G \) with non-trivial automorphisms. Let \( e_1, \ldots, e_k \) be the list of edge-orbits sorted in the order of decreasing weight (the weights \( w \) on all edges in the same orbit by definition must be the same). The main question therefore is how many edges in each edge-orbit \( e_i \) will be added to the spanning tree by the Kruskal’s algorithm. Let \( G_i \) be the subgraph of \( G \) formed by the set of all the edges and nodes in \( e_1, \ldots, e_i \). Let \( V(G_i) \) and \( C(G_i) \) denote the set of nodes and set of connected components of a graph, respectively. Then (see the supplementary material for proof)

**Lemma 4.** The number of edges in \( e_i \) appearing in the MST found by the Kruskal’s algorithm is \( \delta_V^{(i)} - \delta_C^{(i)} \) where \( \delta_V^{(i)} = |V(G_i)| - |V(G_{i-1})| \) and \( \delta_C^{(i)} = |C(G_i)| - |C(G_{i-1})| \). Thus a solution for the linear program (9) is \( \bar{\rho}(e_i) = \frac{\delta_V^{(i)} - \delta_C^{(i)}}{|e_i|} \).

5.2 Lifted Counting of the Number of Connected Components

We note that counting the number of nodes can be done simply by adding the size of each node orbit. The remaining difficulty is how to count the number of connected components of a given graph \( G \) using only information at the orbit level. Let \( \bar{G} \) be the lifted graph of \( G \). Then (see supplementary material for proof)

**Lemma 5.** If \( \bar{G} \) is connected then all connected components of \( G \) are isomorphic. Thus if furthermore \( \bar{G} \) is a connected component of \( G \) then \( |C(G)| = |V(G)|/|V(\bar{G})| \).

To find just one connected component, we can choose an arbitrary node \( u \) and compute \( \bar{G}[u] \), the lifted graph fixing \( u \) (see section 8.1 in [3]), then search for the connected component in \( \bar{G}[u] \) that contains \( \{u\} \). Finally, if \( \bar{G} \) is not connected, we simply apply lemma 5 for each connected component of \( \bar{G} \).

The final lifted Kruskal’s algorithm combines lemma 4 and 5 while keeping track of the set of connected components of \( \bar{G} \) incrementally. The full algorithm is given in the supplementary material.

6 Tightening via Exchangeable Polytope Constraints

One type of symmetry often found in first-order probabilistic models are large sets of exchangeable random variables. In certain cases, exact inference with exchangeable variables is possible via lifted counting elimination and its generalization [17, 2]. The drawback of these exact methods is that they do not apply to many models (e.g., those with transitive clauses). Lifted variational inference methods do not have this drawback, however local and cycle relaxation can be shown to be loose in the exchangeable setting, a potentially serious limitation compared to earlier work.

To remedy this situation, we now show how to take advantage of highly symmetric subset of variables to tighten the relaxation of the lifted marginal polytope.

We call a set of random variables \( \chi \) an exchangeable cluster iff \( \chi \) can be arbitrary permuted while preserving the probability model. Mathematically, the lifting group \( \mathbb{A}_\Delta \) acts on \( \chi \) and the image of the action is isomorphic to \( \mathbb{S}(\chi) \).

\(^8\)Since we are only interested in connectivity in this subsection, the weights of \( G \) play no role. Thus, orbits in this subsection can also be generated by the automorphism group of the unweighted version of \( G \).
the symmetric group on $\chi$. The distribution of the random variables in $\chi$ is also exchangeable in the usual sense.

Our method for tightening the relaxation of the marginal polytope is based on lift-and-project, where we introduce auxiliary variables specifying the joint distribution of a large cluster of variables, and then enforce consistency between the cluster distribution and the pseudo-marginal vector $[20, 24, 27]$. In the ground model, one typically works with small clusters (e.g., triplets) because the number of variables grows exponentially with cluster size. The key (and nice) difference in the lifted case is that we can make use of very large clusters of highly symmetric variables: while the grounded relaxation would clearly blow up, the corresponding lifted relaxation can still remain compact.

Specifically, for an exchangeable cluster $\chi$ of arbitrary size, one can add cluster consistency constraints for the entire cluster and still maintain tractability. To keep the exposition simple, we assume that the variables are binary. Let $\mathcal{C}$ denote a $\chi$-configuration, i.e., a function $\mathcal{C}: \chi \rightarrow \{0, 1\}$. The set $\{\mathcal{C} \mid \forall \text{ configuration } \mathcal{C}\}$ is the collection of $\chi$-cluster auxiliary variables. Since $\chi$ is exchangeable, all nodes in $\chi$ belong to the same node orbit; we call this node orbit $w(\chi)$. Similarly, $e(\chi)$ and $a(\chi)$ denote the single edge and arc orbit that contains all edges and arcs in $\chi$ respectively. Let $u_1, u_2$ be two distinct nodes in $\chi$. To enforce consistency between the cluster $\chi$ and the edge $\{u_1, u_2\}$ in the ground model, we introduce the constraints

$$\sum_{\mathcal{C} \text{ s.t. } \mathcal{C}(u_1) = s_i} \tau_{\mathcal{C}} = \tau_{u_1; s_1, u_2; s_2} \quad \forall s_i \in \{0, 1\} \quad (10)$$

These constraints correspond to using intersection sets of size two, which can be shown to be the exact characterization of the marginal polytope involving variables in $\chi$ if the graphical model only has pairwise potentials. If higher-order potentials are present, a tighter relaxation could be obtained by using larger intersection sets together with the techniques described below.

The constraints in (10) can be methodically lifted by replacing occurrences of ground variables with lifted variables at the orbit level. First observe that in place of the grounded variables $\tau_{u_1; s_1, u_2; s_2}$, the lifted local relaxation has three corresponding lifted variables, $\bar{\tau}_{\chi(00)}, \bar{\tau}_{\chi(11)}$ and $\bar{\tau}_{\chi(01)}$. Second, we consider the orbits of the set of configurations $\mathcal{C}$. Since $\chi$ is exchangeable, there can be only $|\chi| + 1$ $\chi$-configuration orbits; each orbit contains all configurations with precisely $k$ 1’s where $k = 0 \ldots |\chi|$. Thus, instead of the $2^{|\chi|}$ ground auxiliary variables, we only need $|\chi| + 1$ lifted cluster variables. Further manipulation leads to the following set of constraints, which we call lifted exchangeable polytope constraints.

**Theorem 6.** Let $\chi$ be an exchangeable cluster of size $n$; $e(\chi)$ and $a(\chi)$ be the single edge and arc orbit of the graphical model that contains all edges and arcs in $\chi$ respectively; $\bar{\tau}$ be the lifted marginals. Then there exist $c^\chi_k \geq 0, k = 0 \ldots n$ such that

$$\sum_{k=0}^{n-2} \frac{(n-k)(n-k-1)}{n(n-1)} c^\chi_k = \bar{\tau}_{e(\chi):00}$$

$$\sum_{k=0}^{n-2} \frac{(k+1)(k+2)}{n(n-1)} c^\chi_{k+2} = \bar{\tau}_{e(\chi):11}$$

$$\sum_{k=0}^{n-2} \frac{(n-k-1)(n-k+1)}{n(n-1)} c^\chi_{k+1} = \bar{\tau}_{a(\chi):01}$$

**Proof.** See the supplementary material. \hfill $\square$

In contrast to the lifted local and cycle relaxations, the number of variables and constraints in the lifted exchangeable relaxation depends linearly on the domain size of the first-order model. From the lifted local constraints given by [3], $\bar{\tau}_{e(\chi):00} + \bar{\tau}_{e(\chi):11} + 2\bar{\tau}_{a(\chi):01} = 1$. Substituting in the expression involved $c^\chi_k$, we get $\sum_{k=0}^{n} c^\chi_k = 1$. Intuitively, $c^\chi_k$ represents the approximation of the marginal probability $Pr(\sum_{i \in \chi} x_i = k)$ of having precisely $k$ ones in $\chi$.

As proved by [2], groundings of unary predicates in Markov Logic Networks (MLNs) gives rise to exchangeable clusters. Thus, for MLNs, the above theorem immediately suggests a tightening of the relaxation: for every unary predicate of a MLN, add a new set of constraints as above to the existing lifted local (or cycle) optimization problem. Although it is not the focus of our paper, we note that this should also improve the lifted MAP inference results of [3]. For example, in the case of a symmetric complete graphical model, lifted MAP inference using the linear program given by these new constraints would find the exact $k$ that maximizes $Pr(\chi)$, hence recover the same solution as counting elimination. Marginal inference may still be inexact due to the tree-reweighted entropy approximation. We re-emphasize that the complexity of variational inference with lifted exchangeable constraints is guaranteed to be polynomial in the domain size, unlike exact methods based on lifted counting elimination and variable elimination.

### 7 Experiments

In this section, we provide an empirical evaluation of our lifted tree reweighted (LTRW) algorithm. As a baseline we use a dampered version of the lifted belief propagation (LBP-Dampening) algorithm from [21]. Our lifted algorithm has all of the same advantages of the tree-reweighted approach over belief propagation, which we will illustrate in the results: (1) a convex objective that can be convergently solved to optimality, (2) upper bounds on the partition function, and (3) the ability to easily improve the approximation by tightening the relaxation. Our evaluation includes four variants of the LTRW algorithm corresponding to using different outer bounds: lifted local polytope (LTRW-L), lifted cycle polytope (LTRW-C), lifted local
polytope with exchangeable polytope constraints (LTRW-LE), and lifted cycle polytope with exchangeable constraints (LTRW-CE). The conditional gradient optimization of the lifted TRW objective terminates when the duality gap is less than $10^{-4}$ or when a maximum number of 1000 iterations is reached. To solve the LP problem during conditional gradient, we use Gurobi\(^5\).

We evaluate all the algorithms using several first-order probabilistic models. We assume that no evidence has been observed, which results in a large amount of symmetry. Even without evidence, performing marginal inference in first-order probabilistic models can be very useful for maximum likelihood learning [13]. Furthermore, the fact that our lifted tree-reweighted variational approximation provides an upper bound on the partition function enables us to maximize a lower bound on the likelihood [29], which we demonstrate in Sec. 7.5. To find the lifted orbit partition, we use the renaming group as in [3] which exploits the symmetry of the unobserved contents in the model.

Rather than optimize over the spanning tree polytope, which is computationally intensive, most TRW implementations use a single fixed choice of edge appearance probabilities, e.g. an (un)weighted distribution obtained using the matrix-tree theorem. In these experiments, we initialize the lifted edge appearance probabilities $\tilde{\rho}$ to be the most uniform per-orbit edge-appearance probabilities by solving the optimization problem $\inf_{\tilde{\rho} \in \mathcal{T}_{\{|E|\}}} (\tilde{\rho} - \frac{|V|-1}{|E|})^2$ using conditional gradient. Each direction-finding step of this conditional gradient solves a lifted MST problem of the form $\sup_{\tilde{\rho}' \in \mathcal{T}_{\{|E|\}}} \left\langle -2(\tilde{\rho} - \frac{|V|-1}{|E|}), \tilde{\rho}' \right\rangle$ using our lifted Kruskal’s algorithm, where $\tilde{\rho}$ is the current solution. After this initialization, we fix the lifted edge appearance probabilities and do not attempt to optimize them further.

7.1 Test models

Fig. 3 describes the four test models in MLN syntax. We focus on the repulsive case, since for attractive models, all TRW variants and lifted LBP give similar results. The parameter $W$ denotes the weight that will be varying during the experiments. In all models except Clique-Cycle, $W$ acts like the “local field” potential in an Ising model; a negative (or positive) value of $W$ means the corresponding variable tends to be in the 0 (or 1) state. Complete-Graph is equivalent to an Ising model on the complete graph of size $n$ (the domain size) with homogenous parameters. Exact marginals and the log-partition function can be computed in closed form using lifted counting elimination. The weight of the interaction clause is set to $-0.1$ (repulsive). Friends-Smokers (negated) is a variant of the Friends-Smokers model [21] where the weight of the final clause is set to -1.1 (repulsive). We use the method in [2] to compute the exact marginal for the Cancer predicate and the exact value of the log-partition function. Lovers-Smokers is the same MLN used in [3] with a full transitive clause and where we vary the prior of the Loves predicate. Clique-Cycle is a model with 3 cliques and 3 bipartite graphs in between. Its corresponding ground graphical model is shown in Fig. 2.

7.2 Accuracy of Marginals

Fig. 4 shows the marginals computed by all the algorithms as well as exact marginals on the Complete-Graph and Friends-Smokers models. We do not know how to efficiently perform exact inference in the remaining two models, and thus do not measure accuracy for them. The result on complete graphs illustrates the clear benefit of tightening the relaxation: LTRW-Local and LBP are inaccurate for moderate $W$, whereas cycle constraints and, especially, exchangeable constraints drastically improve accuracy. As discussed earlier, for the case of symmetric complete graphical models, the exchangeable constraints suffice to exactly characterize the marginal polytope. As a result, the approximate marginals computed by LTRW-LE and LTRW-CE are almost the same as the exact marginals; the very small difference is due to the entropy approximation. On the Friends-Smokers (negated) model, all LTRW variants give accurate marginals while lifted LBP even with very strong dampening (0.9 weight given to previous iterations’ messages) fails to converge for $W < 2$. We observed that LTRW-LE gives the best trade-off between accuracy and running time for this model. Note that we do not compare to ground versions of the lifted TRW algorithms because, by Theorem 3, the marginals and log-partition function are the same for both.

7.3 Quality of Log-Partition Upper bounds

Fig. 5 plots the values of the upper bounds obtained by the LTRW algorithms on the four test models. The results clearly show the benefits of adding each type of constraint to the LTRW, with the best upper bound obtained by tightening the lifted local polytope with both lifted cycle and exchangeable constraints. For the Complete-Graph and Friends-Smokers model, the log-partition approximation using exchangeable polytope constraints is very close to exact. In addition, we illustrate lifted LBP’s approximation of the log-partition function on the Complete-Graph (note it is non-convex and not an upper bound).

\(^5\)http://www.gurobi.com/
Marginal probability

Clique−Cycle Graph (Domain Size = 100)
Friends−Smokers (Domain Size = 200)

ent iterations, thus is faster; however note that its running
magnitudes faster than the ground version. Interestingly,
As shown in Table 1, lifted variants of TRW are order-of-

"Lovers-Smokers"

W

\[ x \neq y \land Loves(x, y) \]

0.5

\[ x \neq y \land Male(x) \land Smokes(x) \]

1

\[ x \neq y \land Loves(x, y) \land (Smokes(x) \Rightarrow Smokes(y)) \]

−100

\[ x \neq y \land y \neq z \land z \neq x \land Loves(x, y) \land Loves(y, z) \land Loves(x, z) \]


\[ W \]

\[ x \neq y \land (Q1(x) \Rightarrow Q1(y)) \]

\[ x \neq y \land (Q2(x) \Rightarrow Q3(y)) \]

\[ x \neq y \land (Q3(x) \Rightarrow Q4(y)) \]

\[ x \neq y \land (Q4(x) \Rightarrow Q2(y)) \]

Figure 3: Test models

Figure 5: Approximations of the log-partition function on the four test models from Fig. 3 (best viewed in color).

Figure 4: Left: marginal accuracy for complete graph model. Right: marginal accuracy for \( P_r(Cancer(x)) \) in Friends-Smokers (neg). Lifted TRW variants using different outer bounds: L=local, C=cycle, LE=local+exchangeable, CE=cycle+exchangeable (best viewed in color).

7.4 Running time

As shown in Table 1, lifted variants of TRW are order-of-magnitudes faster than the ground version. Interestingly, lifted TRW with local constraints is observed to be faster as the domain size increase; this is probably due to the fact that as the domain size increases, the distribution becomes more peak, so marginal inference becomes more similar to MAP inference. Lifted TRW with local and exchangeable constraints requires a smaller number of conditional gradient iterations, thus is faster; however note that its running time slowly increases since the exchangeable constraint set grows linearly with domain size.

LBP’s lack of convergence makes it difficult to have a meaningful timing comparison with LBP. For example, LBP did not converge for about half of the values of \( W \) in the Lovers-Smokers model, even after using very strong dampening. We did observe that when LBP converges, it is much faster than LTRW. We hypothesize that this is due to the message passing nature of LBP, which is based on a fixed point update whereas our algorithm is based on Frank-Wolfe.

7.5 Application to Learning

We now describe an application of our algorithm to the task of learning relational Markov networks for inferring protein-protein interactions from noisy, high-throughput, experimental assays [12]. This is equivalent to learning the parameters of an exponential family random graph model [19] where edges in the random graph represent the protein-protein interactions. Despite fully observed data, maximum likelihood learning is challenging because of the intractability of computing the log-partition function and its gradient. In particular, this relational Markov network has over 330K random variables (one for each possible interaction of 813 variables) and tertiary potentials. However, Jaimovich et al. [13] observed that the partition function in
relational Markov networks is highly symmetric, and use lifted LBP to efficiently perform approximate learning in running time that is independent of the domain size. They use their lifted inference algorithm to visualize the (approximate) likelihood landscape for different values of the parameters, which among other uses characterizes the robustness of the model to parameter changes.

We use precisely the same procedure as [13], substituting lifted BP with our new lifted TRW algorithms. The model has three parameters: $\theta_1$, used in the single-node potential to specify the prior probability of a protein-protein interaction; $\theta_{011}$, part of the tertiary potentials which encourages cliques of three interacting proteins; and $\theta_{001}$, also part of the tertiary potentials which encourages chain-like structures where proteins $A$, $B$ interact, $B$, $C$ interact, but $A$ and $C$ do not (see supplementary material for the full model specification as an MLN). We follow their two-step estimation procedure, first estimating $\theta_1$ in the absence of the other parameters (the maximum likelihood, BP, and TRW estimates of this parameter coincide, and estimation can be performed in closed-form: $\theta_1^* = -5.293$). Next, for each setting of $\theta_{011}$ and $\theta_{001}$ we estimate the log-partition function using lifted TRW with the cycle+exchangeable vs. local constraints only. Since TRW is an upper bound on the log-partition function, these provide lower bounds on the likelihood.

Our results are shown in Fig. 6, and should be compared to Fig. 7 of [13]. The overall shape of the likelihood landscapes are similar. However, the lifted LBP estimates of the likelihood have several local optima, which cause gradient-based learning with lifted LBP to reach different solutions depending on the initial setting of the parameters. In contrast, since TRW is convex, any gradient-based procedure would reach the global optima, and thus learning is much easier. Interestingly, we see that our estimates of the likelihood have a significantly smaller range over these parameter settings than that estimated by lifted LBP. For the lifted TRW algorithms there is a sudden decrease in the likelihood at $\theta_{011} > 0$ and $\theta_{111} > 0.02$ (so as to not affect the scale, we show this as white).

8 Discussion and Conclusion

Lifting partitions used by lifted and counting BP [21, 16] can be coarser than orbit partitions. In graph-theoretic terms, these partitions are called equitable partitions. If each equitable partition cell is thought of as a distinct node color, then among nodes with the same color, their neighbors must have the same color histogram. It is known that orbit partitions are always equitable, however the converse is not always true [9].

Since equitable partition can be computed more efficiently and potentially leads to more compact lifted problems, the following question naturally arises: can we use equitable partition in lifting the TRW problem? Unfortunately, a complete answer is non-trivial. We point out here a theoretical barrier due to the interplay between the spanning tree polytope and the equitable partition of a graph.

Let $\varepsilon$ be the coarsest equitable partition of edges of $G$. We give an example graph in the supplementary material (see example 9) where the symmetrized spanning tree polytope corresponding to the equitable partition $\varepsilon$, $T_\varepsilon \cap B_\varepsilon^{(E)}$ is an empty set. When $T_\varepsilon$ is empty, the consequence is that if we want $\rho$ to be within $\mathbb{T}$ so that $B(\ldots, \rho)$ is guaranteed to be a convex upper bound of the log-partition function, we cannot restrict $\rho$ to be consistent with the equitable partition. In lifted and counting BP, $\rho \equiv 1$ so it is clearly consistent with the equitable partition; however, one loses convexity and upper bound guarantee as a result. This suggests that there might be a trade-off between the compactness of the lifting partition and the quality of the entropy approximation, a topic deserving the attention of future work.

In summary, we presented a formalization of lifted marginal inference as a convex optimization problem and showed that it can be efficiently solved using a Frank-Wolfe algorithm. Compared to previous lifted variational inference algorithms, in particular lifting belief propagation, our approach comes with convergence guarantees, upper bounds on the partition function, and the ability to improve the approximation (e.g. by introducing additional constraints) at the cost of small additional running time.

A limitation of our lifting method is that as the amount of soft evidence (the number of distinct individual objects) approaches the domain size, the behavior of lifted inference approaches ground inference. The wide difference in running time between ground and lifted inference suggests that significant efficiency can be gained by solving an approximation of the original problem that is more symmetric [25, 15, 22, 6]. One of the most interesting open questions raised by our work is how to use the variational formulation to perform approximate lifting. Since our lifted TRW algorithm provides an upper bound on the partition function, it is possible that one could use the upper bound to guide the choice of approximation when deciding how to re-introduce symmetry into an inference task.

Acknowledgements: Work by DS supported by DARPA PPAML program under AFRL contract no. FA8750-14-C-0005.
References


Supplementary Materials for “Lifted Tree-Reweighted Variational Inference”

We present (1) proofs not given in the main paper, (2) full pseudo-code for the lifted Kruskal’s algorithm for finding maximum spanning tree in symmetric graphs, and (3) additional details of the protein-protein interaction model.

Proof of Theorem 1.

Proof. The lifting group \( \mathcal{A}_{\Delta} \) stabilizes both the objective function and the constraints of the convex optimization problem in the LHS of Eq. (4). The equality is then established using Lemma 1 in [3]. \( \square \)

We state and prove a lemma about the symmetry of the bounds \( B \) and \( B^* \) that will be used in subsequent proofs.

Lemma 7. Let \( \pi \) be an automorphism of the graphical model \( \mathcal{G} \), then \( B^*(\tau, \rho) = B^*(\pi^* \tau, \rho^*) \) and \( B(\theta, \rho) = B(\theta^*, \rho^*) \).

Proof. The intuition is that since the entropy bound \( B^* \) is defined on the graph structure of the graphical model \( \mathcal{G} \), it inherits the symmetry of \( \mathcal{G} \). This can be verified by viewing the graph automorphism \( \pi \) as a bijection from nodes to nodes and edges to edges, and so \( B^*(\pi^* \tau, \rho^*) \) simply rearranges the summation inside \( B^*(\tau, \rho) \).

Thus \( B^*(\pi^* \tau, \rho^*) = B^*(\pi^* \tau, \rho^*) \). On the other hand, by Lemma 7, \( B^*(\pi^* \tau, \rho^*) = B^*(\tau, \rho) \). Thus \( B^*(\tau, \rho) = B^*(\tau, \rho) \). Note that in case of the overcomplete representation, the action of the group \( \mathcal{A}_{\Delta} \) is the permuting action of \( \pi \); thus, the TRW bound \( B^*(\tau, \rho) \) (for fixed \( \rho \in T_{[\rho, E]} \)) is stabilized by the lifting group \( \mathcal{A}_{\Delta} \). \( \square \)

Conditional Gradient (Frank-Wolfe) Algorithm for Lifted TRW

The pseudo-code is given in Algorithm 1. Step 2 essentially solves a lifted MAP problem which we used the same algorithms presented in [3] with Gurobi as the main linear programming engine. Step 3 solves a 1-D constrained convex problem via the golden search algorithm to find the optimal step size.

Proof of Lemma 4.

Proof. After considering all edges in \( e_1 \cup \ldots \cup e_i \), Kruskal’s algorithm must form a spanning forest of \( \mathcal{G}_i \) (the forest is spanning since if there remains an edge that can be used without forming a cycle, Kruskal’s algorithm must have used it already). Since the forest is spanning, the number of edges used by Kruskal’s algorithm at this point is precisely \( |V(\mathcal{G}_i)\mathcal{C}(\mathcal{G}_i)| \). Similarly, just after considering all edges in \( e_i \cup \ldots \cup e_{i-1} \), the number of edges used is \( |V(\mathcal{G}_i)| - |C(\mathcal{G}_i)\mathcal{C}(\mathcal{G}_i)| \). Therefore, the number of \( e_i \)-edges used must be

\[
|V(\mathcal{G}_i)| - |C(\mathcal{G}_i)| - |V(\mathcal{G}_i)| - |C(\mathcal{G}_i)| = \delta_V^i - \delta_C^i
\]

which is the difference between the number of new nodes (which must be non-negative) and the number of new connected components (which could be negative) induced by considering edges in \( e_i \). Any MST solution \( \rho \) can be turned into a solution \( \hat{\rho} \) of (9) by letting \( \hat{\rho}(e) = \frac{1}{|e|} \sum_{e' \in e} \rho(e') \).

Thus, we obtain a solution \( \hat{\rho}(e_i) = \frac{\delta_V^i - \delta_C^i}{|e_i|} \).

Proof of Lemma 5.

We first need an intermediate result.

Lemma 8. If \( u \) and \( v \) are two distinct node orbits, and \( u \) and \( v \) are reachable in the lifted graph \( \mathcal{G} \), then for any \( u \in \mathcal{U} \), there is some \( v \in \mathcal{V} \) such that \( v \) is reachable from \( u \).

Proof. Induction on the length of the \( u \)-\( v \) path. Base case: if \( u \) and \( v \) are adjacent, there exists an edge orbit \( e \) incident to both \( u \) and \( v \). Therefore, the exists a ground edge \( \{u_0, v_0\} \in e \) such that \( u_0 \in u \) and \( v_0 \in v \). The automorphism mapping \( u_0 \rightarrow u \) will map \( v_0 \rightarrow v \) for some node \( v \). Clearly \( \{u, v\} \) is an edge, and \( v \in \mathcal{V} \). Main case: assume the statement is true for all pair of orbits with path length \( \leq n \). Suppose \( u \rightarrow v \) is a path of length \( n + 1 \). Take the orbit \( z \) right in front of \( v \) in the path, so that \( u \rightarrow z \) is a path of length \( n \), and \( z \) and \( v \) are adjacent. By the inductive assumption, there exists \( z \in \mathcal{Z} \) such that \( u \) is connected to \( z \).
Applying the same argument of the base case, there exists \( v \in V \) such that \( \{ z, v \} \) is an edge. Thus \( u \) is connected to \( v \).

We now return to the main proof of lemma 5.

**Proof.** If the ground graph \( G \) has only one component then this is trivially true. Let \( G_1 \) and \( G_2 \) be two distinct connected components of \( G \), let \( u_1 \) be a node in \( G_1 \) and \( u \) be the orbit containing \( u_1 \). Let \( v_2 \) be any node in \( G_2 \). Since the lifted graph \( \tilde{G} \) is connected, all orbits are reachable from one another in \( \tilde{G} \). By the above lemma, there must be some node \( u_2 \) reachable from \( v_2 \), hence \( u_2 \in G_2 \) (if \( v_2 \in u \) then we just take \( u_2 = v_2 \)). This establishes that the node orbit \( u \) intersects with both \( G_1 \) and \( G_2 \). Note that \( u_1 \neq u_2 \) since otherwise \( G_1 = G_2 \). Let \( \pi \) be the automorphism that takes \( u_1 \) to \( u_2 \).

We now show that \( \pi(G_1) = G_2 \). Since \( \pi \) maps edges to edges and non-edges to non-edges, it is sufficient to show that \( \pi(V(G_1)) = \pi(V(G_2)) \). Let \( z_1 \) be a node of \( G_1 \) and \( z_2 = \pi(z_1) \). Since \( G_1 \) is connected, there exists a path from \( u_1 \) to \( z_1 \). But \( \pi \) must map this path to a path from \( u_2 \) to \( z_2 \) since \( u_2 \) and \( z_2 \) are in the same orbit. By a similar argument, \( \pi^{-1} \) must map the path from \( u_2 \) to \( z_2 \) to a path from \( u_1 \) to \( z_1 \). Hence all connected components of \( G \) are isomorphic. Given one connected component \( G' \), the number of connected components of \( G \) is \( |C(G)| = |V(G')|/|V(G')| \).

**Lifted Kruskal’s Algorithm**

See algorithm 2. The algorithm keeps track of the list of connected components of the lifted graph \( \tilde{G} \) in a disjoint-set data structure similar to Kruskal’s algorithm. Line 17 and line 20 follow from lemma 5 and lemma 4 respectively. The number of ground nodes of a lifted graph is computed as the sum of the size of each of its node orbit.

**Proof of Theorem 6.**

We give two different proofs of this theorem. The first proof demonstrates how the constraints are derived methodically by lifting the ground constraints of the exact marginal polytope. The second proof is more intuitive and illustrates what each variable in the system of constraints represents conceptually.

**Proof 1.**

First consider the configuration \( \mathcal{C} \) such that \( \mathcal{C}(u_1, u_2) = (0, 0) \). From Eq. (10), after substituting the ground variables \( \tau^k_\mathcal{C} \) by the lifted variables \( \bar{\tau}^{k}_{|\mathcal{C}|} \), we obtain the lifted constraint

\[
\exists \hat{\tau}_0, \ldots, \hat{\tau}_n : \sum_{\mathcal{C}} \hat{\tau}^X_{|\mathcal{C}|} = \bar{\tau}_{e(\cdot):00}
\]

Let us now simplify the summation. Since \( \mathcal{C}(u_1, u_2) = (0, 0) \), \( |\mathcal{C}| \) can range from 0 to \( n - 2 \). For each value of \( |\mathcal{C}| = k \), there are \( \binom{n-2}{k} \) different configurations. As a result, we can compactly write the above lifted constraint as

\[
\sum_{k=0}^{n-2} \binom{n-2}{k} \bar{\tau}^X_k = \bar{\tau}_{e(\cdot):00}
\]

Note that every edge \( \{ u_1, u_2 \} \) results in exactly the same constraint.

Similarly, when \( \mathcal{C}(u_1, u_2) = (1, 1) \) we obtain the lifted constraint

\[
\sum_{k=0}^{n-2} \binom{n-2}{k} \bar{\tau}^X_{k+2} = \bar{\tau}_{e(\cdot):11}
\]

and when \( \mathcal{C}(u_1, u_2) = (0, 1) \) we obtain

\[
\sum_{k=0}^{n-2} \binom{n-2}{k} \bar{\tau}^X_{k+1} = \bar{\tau}_{e(\cdot):01}
\]

There’s no need to consider the \((1, 0)\) case since we can always re-order the pair \((u_1, u_2)\).
Finally, let $c^χ_k = \binom{k}{i} \hat{r}^χ_k$, we arrive at the set of constraints of theorem 6. \hfill\Box

Proof 2.

Proof. Let $C(0, 0)$ denote the number of $(0, 0)$ edges in $\chi$. An $(0, 0)$ edge is an edge where the two end-nodes receive the assignment 0. To show the first equality holds, we use two different ways to compute the expectation of $C(0, 0)$.

First, $\mathbb{E}[C(0, 0)]$ is the sum of the probability that an edge is $(0, 0)$, summing over all the edges. Due to exchangeability, all probabilities are the same and is equal to $\hat{r}_{e(\chi):00}$, so

$$\mathbb{E}[C(0, 0)] = \frac{n(n-1)}{2} \hat{r}_{e(\chi):00}$$

Second, $C(0, 0)$ conditioned on the event that there are precisely $k$ 1’s in $\chi$ is $\binom{n-k}{2} \hat{r}^χ_{e(\chi):00}$ for $k \leq n-2$ and 0 if $k > n-2$. Now let $c^χ_k$ be the probability of this event, then

$\mathbb{E}[C(0, 0)] = \mathbb{E}[\mathbb{E}[C(0, 0) | \sum x_i = k]]$

$= \sum_{k=0}^{n-2} \frac{(n-k)(n-k-1)}{2} c^χ_k$

This shows the first equality holds. The second equality is obtained similarly by considering the expectation of $C(1, 1)$, the number of $(1, 1)$ edges in $\chi$. The last equality is obtained by considering the expectation of $C(0, 1)$, the number of $(0, 1)$ arcs. \hfill\Box

Protein-Protein Interaction (PPI) Model

The PPI model we considered is exactly the same as an exponential family random graph model with 3 graph statistics: edge count, 2-chain (triangle with a missing edge) count, and triangle count. The model specification in MLN (with an additional clause enforcing undirectedness of the random graph) is

$$\frac{1}{2} \theta_1 \quad r(x, y)$$

$$\frac{1}{2} \theta_{10} \quad r(x, y) \land r(x, z) \land r(y, z)$$

$$\frac{1}{6} \theta_{111} \quad r(x, y) \land r(y, z) \land r(x, z)$$

$$-\infty \quad !(r(x, y) \leftrightarrow r(y, x))$$

For this model, the edge appearance $\hat{\rho}$ is initialized so that the edges correspond to the hard clause are always selected.
The set \( \{ r(x,y) \mid x \text{ fixed, } y \text{ varies} \} \) is an exchangeable cluster, thus exchangeable constraints can be enforced on the orbit corresponding to this cluster. The log-partition upper bounds of lifted TRW with different outer bounds are shown in Fig. 7. The left part shows the region with negative \( \theta_{011} \) and the right part shows the region with positive \( \theta_{011} \) (\( \theta_{111} \) is held fixed at \(-0.021\)). For negative \( \theta_{011} \), a combination of cycle and exchangeable constraints (CE) is crucial to improve the upper bound. For positive \( \theta_{011} \), exchange constraints (LE) are already sufficient to yield the best bound.

**Equitable Partition and the Spanning Tree Polytope**

**Example 9.** We give an example where the symmetrized spanning tree polytope corresponding to the (edge) equitable partition \( \varepsilon \), \( T[\varepsilon] = T(G) \cap R_{[\varepsilon]}[E] \) is an empty set. The example graph \( G \) is shown in Fig. 8. There are two edge types in the coarsest equitable partition: solid and dashed. The dashed edge in the middle is a bridge, it must appear in every spanning tree, so every \( \rho \in T \) must assign 1 to this edge. If \( T[\varepsilon] \neq \emptyset \) then there is some \( \rho \in T[\varepsilon] \) that assigns the same weights to all dashed edges. Therefore it will assign 1 to the two dashed edges on the left and right hand side. The remaining solid edges have equal weight, and since the total weight is \( |V| - 1 = 7 \), the solid weight is \((7 - 3)/6 = 2/3\). Now consider the triangle on the left. This triangle has the total weight \( 1 + 4/3 > 2 \) which violates the constraint of the spanning tree polytope. Thus, for this graph \( T[\varepsilon] \) must be empty.