

Lifted Message Passing for Hybrid Probabilistic Inference

Yuqiao Chen*, Nicholas Ruozzi and Sriraam Natarajan

University of Texas at Dallas

{yuqiao.chen, nicholas.ruozzi, sriraam.natarajan}@utdallas.edu

Abstract

Lifted inference algorithms for first-order logic models, e.g., Markov logic networks (MLNs), have been of significant interest in recent years. Lifted inference methods exploit model symmetries in order to reduce the size of the model and, consequently, the computational cost of inference. In this work, we consider the problem of lifted inference in MLNs with continuous or both discrete and continuous groundings. Existing work on lifting with continuous groundings has mostly been limited to special classes of models, e.g., Gaussian models, for which variable elimination or message-passing updates can be computed exactly. Here, we develop approximate lifted inference schemes based on particle sampling. We demonstrate empirically that our approximate lifting schemes perform comparably to existing state-of-the-art models for Gaussian MLNs, while having the flexibility to be applied to models with arbitrary potential functions.

1 Introduction

Statistical Relational Learning/AI (SRL) [Getoor and Taskar, 2007; Raedt *et al.*, 2016] models combine the representational power of logic/relational models with the ability of probability theory to model noise and uncertainty. While powerful in representation, until recently, learning and reasoning in these models were considered challenging. On the reasoning side, given a first-order (logical) model, a class of probabilistic inference methods called *lifted inference* [Poole, 2003] methods were developed to perform efficient inference. The key idea is to exploit the underlying symmetries that exist in the model (top-down) or the data (bottom-up). The top-down approaches [Poole, 2003; de Salvo Braz *et al.*, 2005; den Broeck, 2013; Gogate and Domingos, 2011] perform splitting or shattering where the groups of similar objects are split according to the evidence. The bottom-up approaches [Kersting *et al.*, 2009] start from a fully grounded model and group variables based on the evidence. Similar to standard graphical model inference, there exist exact and approximate methods. Ex-

act lifted inference is intractable in the presence of evidence while message passing [Singla and Domingos, 2008; Kersting *et al.*, 2009] and sampling [Venugopal and Gogate, 2012; Niépert, 2012] can handle evidence more robustly.

However, these methods have been mainly applied to discrete domains, with the exception of Lifted Gaussian BP [Ahmadi *et al.*, 2011] and Lifted Kalman Filters [Choi *et al.*, 2011] which apply to continuous models with a Gaussian distribution assumption, and do not extend to hybrid models.

Inspired by the success of color passing [Kersting *et al.*, 2009], we develop a particle message-passing approach that passes messages between clustered variables and factors. We demonstrate how the proposal distributions need to be adapted to the clusters by counting the number of objects in each cluster. In addition, combining the idea of coarse-to-fine [Habeeb *et al.*, 2017; Sarkhel *et al.*, 2015] and evidence grouping [Venugopal and Gogate, 2014], we derive a coarse-to-fine algorithm that computes approximate symmetries. Note that exact symmetry computation is quite impractical for continuous domains. Our approach clusters all the evidence in the beginning and, over the course of message passing, refines these clusters to compute more accurate evidence as necessary. Unlike existing approaches to lifting, our method is not limited to simple classes of potential functions, e.g., Gaussian distributions, nor does it require that all variables be discrete or that all variables be continuous. This makes our approach significantly more versatile.

We make these key contributions: (1) We develop the *first hybrid lifted inference algorithm* for real world data. (2) Ours is a generalized lifted inference approach for hybrid data that does not make any distributional assumptions. Our work can be seen as generalizing the work on Lifted Kalman Filters [Choi *et al.*, 2011] and Hybrid MLNs [Wang and Domingos, 2008] to arbitrary (integrable) potentials. (3) We show theoretically that our approach is indeed a reasonable approximation under local conditions. (4) We demonstrate the efficiency and effectiveness of this approach on several tasks against standard baselines where most lifted inference techniques cannot be naturally adapted.

2 Preliminaries

A Markov random field (MRF) is a graphical model defined by a hypergraph $G = (V, C)$ that represents a joint probability distribution of random variables $x_i \in X_i$ for each $i \in V$ as

*Contact Author

a product of nonnegative potential functions over hyperedges of G . Here, \mathbb{X}_i is the set of possible states of the variable i , e.g., the finite sets $\{0, 1\}$ or \mathbb{R} . In this work, we will consider models with a combination of both discrete and continuous variables. Given a collection of nonnegative potential functions $f_c : \mathbb{X}_c \rightarrow \mathbb{R}_{\geq 0}$ for each $c \in C$, the joint probability distribution over $\bigcup_{i \in V} \mathbb{X}_i$ is given by

$$p(x) = \frac{1}{Z} \prod_{c \in C} f_c(x_c),$$

where x_c is the subset of random variables indexed by elements of the clique c and $Z = \prod_{c \in C} f_c(x_c)$ is a normalization constant. If some of the variables are discrete, the integral over those variables is replaced with the appropriate sum, but for ease of notation, we will use the integral notation in what follows. We can represent an MRF with the above factorization using a factor graph, a bipartite graph with variable nodes for each $i \in V$, factor nodes for each $c \in C$, and an edge joining $c \in C$ to $i \in V$ if $i \in c$.

2.1 Loopy BP and Particle Message Passing

Our primary goal is to compute the probability of a set of *query* variables conditioned on *evidence* variables. While this inference task is #P-complete in general, for tree-structured factor graphs, these conditional probabilities can be computed exactly in polynomial time by a message-passing algorithm known as belief propagation (BP). When the factor graph is not tree structured, belief propagation is no longer exact. However, loopy BP has been shown to have reasonable empirical performance as an approximate inference algorithm [Berrou *et al.*, 1993; Murphy *et al.*, 1999].

In the case that all of the variables are discrete, the integrals in the BP message updates reduce to sums and can be computed efficiently. Similarly, if all of the variables are continuous and all of the potential functions are Gaussian, then the message passing algorithm is known as Gaussian belief propagation (GaBP) and the integrals can be computed in closed form [Weiss and Freeman, 2000; Malioutov *et al.*, 2006; Moallemi and Roy, 2009; Ruozzi and Tatikonda, 2013]. Beyond these special cases, however, inference with BP becomes practically challenging: in the case of hybrid models, i.e., models with both continuous and discrete variables, and general potential functions, the messages produced by BP do not necessarily correspond to a finitely parametrizable family of functions nor can the integrals typically be computed in closed form.

Several alternative message-passing schemes have been proposed to address these issues. Nonparametric belief propagation (NBP) represents each message as a mixture of Gaussians and uses an efficient sampler to compute approximate message updates, but this approach is often too slow to be practical [Sudderth *et al.*, 2010]. Another alternative, the particle belief propagation (PBP) method approximates the messages using a finite collection of particles rather than a Gaussian mixture [Ihler and McAllester, 2009]. In this approach, the integrals are approximated via importance sampling or an MCMC sampling method. Specifically, consider a proposal distribution $q_i(x_i)$ for each $i \in V$. In each PBP iteration,

for each $i \in V$, T samples, $x_i^{(1)}, \dots, x_i^{(T)}$, are drawn from the proposal distribution q_i . The PBP approximated message from factor node c to variable node i is defined as

$$\hat{m}_{c \rightarrow i}(x_i) \triangleq \frac{1}{T} \sum_{t=1}^T f(x_i, x_{c \setminus i}^{(t)}) w_{j \rightarrow c}(x_j^{(t)}),$$

where $w_{i \rightarrow c}$ is the importance weighted message from variable node i to factor node c

$$w_{i \rightarrow c}(x_i) \triangleq \frac{1}{T \cdot q(x_i)} \sum_{c \in nb(i) \setminus i} \hat{m}_{c \rightarrow i}(x_i).$$

Upon convergence, the beliefs are computed as

$$b(x_i) \triangleq \frac{1}{c \in nb(i)} \hat{m}_{c \rightarrow i}(x_i). \quad (1)$$

The choice of proposal distribution greatly impacts the practical performance of PBP. Ihler and McAllester [2009] suggest using the approximate beliefs as proposal distributions and drawing samples from them via a standard MCMC sampling approach. An alternative approach that empirically results in faster convergence, is to generate an approximate proposal distribution from some tractable exponential family using a parallel message-passing strategy [Lienart *et al.*, 2015]. This approach is dubbed expectation particle belief propagation (EPBP) as the approximate proposal messages are formed using an approach similar to that of the expectation propagation algorithm [Minka, 2001]. Here, like Lienart *et al.* [2015], we use Gaussian distributions as the tractable exponential family.

2.2 Relational Models

SRL models [Raedt *et al.*, 2016] combine the power of graphical models to model uncertainty and the ability of first-order logic to handle relational data. Examples include the directed Probabilistic Relational Model [Koller, 1999] and undirected Markov Logic Networks [Richardson and Domingos, 2006]. While specific semantics differ, they employ parameterized (conditional probabilities or weights) logic/relational rules. For the purposes of lifted inference, most of these models are converted to parameterized factor graphs [Poole, 2003].

2.3 Color Passing

Lifted message passing methods typically employ the notion of a compressed factor graph G composed of super variables, V , and super factors, C . These sets correspond to a regrouping of the original variable nodes and factor nodes for which the message-passing algorithm behaves identically, i.e., nodes/factors in a cluster send and receive the same messages. While top-down [Singla and Domingos, 2008] and bottom-up [Kersting *et al.*, 2009] methods exist, we employ the later to construct the super variables and factors.

The bottom-up method operates in two key steps - first lifted graph is constructed from a ground probabilistic network and then a modified BP is run on this graph. To construct the lifted graph, the method performs color (structure) passing. Initially, all variable and factor nodes are clustered based on evidence and the potential functions. Variables with

the same evidence value v or variables with same domains will be grouped together, i.e., receive the same color. Each node sends its color to its neighbors. Each factor stacks the colors of all its neighbors, appends its own color, and sends a message back to the neighbors. After each step, the nodes and factors are grouped together based on the colors. The process is repeated until no more grouping is possible at which point, all nodes (correspondingly factors) inside the same cluster send and receive the same messages.

In the second step, this method runs the modified BP by introducing counts – $\#(C, i)$ denotes the number of identical messages that would be sent from the super factor C to each node in the super variable i if BP was carried out on the original factor graph G . The message from a super variable i to a super factor C in the compressed graph is now,

$$m_{i \rightarrow C}(x_i) = \frac{\prod_{c \in nb(i)} m_{c \rightarrow i}(x_i)^{\#(c, i)}}{m_{C \rightarrow i}(x_i)}.$$

The denominator shows that a super variable's message to a super factor excludes the corresponding factor's message to the variable if BP was performed on G . The unnormalized belief of the random variable x_i can be computed by

$$b(x_i) = \prod_{c \in nb(i)} m_{c \rightarrow i}(x_i)^{\#(c, i)}.$$

3 Hybrid Lifted BP

While PBP/EPBP can be applied to general factor graphs with both continuous and discrete variables, care must be taken if these approaches are to be used in a lifted context as lifting with continuous domains introduces some additional challenges that are not present in the fully discrete case. In particular, exploiting symmetries in a factor graph with continuous evidence is complicated by the observation that *each different evidence variable would likely be considered as distinct under the standard color passing algorithm, potentially destroying many of the symmetries in the model*. In this section, we propose an intuitive strategy to generalize the color passing and EPBP algorithms to handle the case of continuous variables in a lifted setting.

Recall that the lifted approximate inference method has two steps - constructing a compressed factor graph from a large grounded graph and then running a modified lifted inference algorithm on this graph. We first show how EPBP can be applied on the compressed graph before outlining the process of constructing the compressed graph.

3.1 Lifted EPBP

We now introduce our Lifted EPBP (LEPBP) algorithm. Consider a compressed graph G with a set of super variables V , a set of super factors C , and proposal distribution $q_i(x_i)$ for each super variable $i \in V$. For each iteration of EPBP, we draw T samples $x_i^{(1)}, \dots, x_i^{(T)}$ for each super variable $i \in V$ from the corresponding proposal distribution $q_i(x_i)$. The *particle approximated message from superfactor C to continuous super variable $i \in C$ is*

$$\hat{m}_{C \rightarrow i}(x_i) \triangleq \prod_{t=1}^T f_C(x_i, x_{C \setminus i}^{(t)}) \prod_{j \in C \setminus i} \hat{w}_{j \rightarrow i}(x_j^{(t)}), \quad (2)$$

Algorithm 1 Lifted Hybrid EPBP

```

1: Input: A factor graph  $G$  with variables  $X$ , factors  $f$ , and evidences  $e$ 
2: Return: The approximated belief  $b(x_i)$ 
3: Run CP until convergence to compute  $G = (V, C)$ 
4: Initialize messages  $\hat{m}$  and proposals  $q$ 
5: repeat
6:   for  $i \in V, c \in nb(i)$  do
7:     Update  $\hat{m}_{i \rightarrow c}(x_i)$ 
8:   end for
9:   for  $c \in C, i \in c$  do
10:    Compute the cavity distribution  $q_{\setminus c}$  from (4)
11:    Update  $\eta_{c \rightarrow i}$  using (5)
12:   end for
13:   for  $i \in V$  do
14:     Sample  $T$  points,  $x_i^{(1)}, \dots, x_i^{(T)}$ , from  $q_i$ 
15:   end for
16:   for  $c \in C, i \in c$  do
17:     Update  $\hat{m}_{c \rightarrow i}(x_i)$  using (2) and (3)
18:   end for
19: until Converge
20: Compute beliefs from the converged messages

```

where \hat{w} is the collection of weighted messages from super variables to super factors

$$\hat{w}_{i \rightarrow c}(x_i) \triangleq \frac{\prod_{c' \in nb(i)} \hat{m}_{c' \rightarrow i}(x_i)^{\#(c', i)}}{T \cdot q_i(x_i) \cdot \hat{m}_{C \rightarrow i}(x_i)}. \quad (3)$$

The weighted messages for a discrete super variable can be computed exactly as in the lifted version of BP.

The proposal distributions, $q_i(x_i)$, used by the lifted EPBP algorithm are computed by introducing the counts into the appropriate place in the standard EPBP algorithm. Define

$$q_i(x_i) \triangleq \prod_{c \in nb(i)} \eta_{c \rightarrow i}(x_i)^{\#(c, i)}, \quad (4)$$

where the η 's represent the approximate messages in the desired exponential family. We define the lifted cavity distribution $q_{\setminus c}(x_i) = q_i(x_i) / \eta_{c \rightarrow i}(x_i)$ and the corresponding tilted distribution $\hat{m}_{c \rightarrow i}(x_i)q_{\setminus c}(x_i)$. The proposal message $\eta_{c \rightarrow i}$ is updated by minimizing the KL-divergence between $q_i(x_i)$ and the tilted distribution.

$$\eta_{c \rightarrow i} = \arg \min_{\eta \in \text{exp.fam.}} D(\hat{m}_{c \rightarrow i}(x_i)q_{\setminus c}(x_i) \parallel \eta(x_i)q_{\setminus c}(x_i)) \quad (5)$$

The complete procedure is described in Algorithm 1. Note that in line 3, it calls a *standard color passing* algorithm that constructs the compressed graph. However, as we mentioned earlier, continuous domains, do not exhibit exact symmetries and hence we need an improved color passing algorithm. We refer to this as coarse-to-fine lifting, which we explain next.

3.2 Coarse-to-fine Lifting

In situations with a large number of continuous evidence variables, the number of exact symmetries in the factor graph is severely limited. Here, we propose a coarse to fine lifting approach for detecting approximate symmetries. Our approach

is to begin with a coarse clustering of the evidence values into k clusters, S_1, \dots, S_k (k could be as small as one). In EPBP, for each $a \in \{1, \dots, k\}$, we will treat all of the evidence variables in S_a as having the same value, given by the cluster average. We can use this coarse clustering as an initial grouping of nodes in a color passing algorithm.

The above clustering scheme introduces an approximation to the evidence: The closer each evidence variable is to the respective mean of its assigned cluster, the more accurate the approximation. However, higher accuracy typically results in fewer model symmetries to exploit. In coarse-to-fine lifting, the aim is to gradually improve the clustering, interleaving it with the lifted EPBP algorithm, while reducing the total computational cost, possibly at the expense of some inaccuracy in the early rounds of LEPBP.

After an initial coarse grouping of the evidence variables, we iteratively refine the grouping of evidence by subdividing the current set of coarse clusters. In this work, we use the k-means algorithm for finding an appropriate split, e.g.,

running k -means for cluster S_a with $k = 2$ will replace the current cluster with two new clusters. Note that this is a one-dimensional k -means clustering application, which can be solved optimally in polynomial time [Wang and Song, 2011]. After splitting the coarse clusters, we run one iteration of color passing and lifted EPBP, using the coarse EPBP messages from last iteration. We can keep iterating this procedure until lifted EPBP converges. In practice, it is not necessary or even desirable to continue splitting evidence groups until each cluster only contains a single evidence variable. We can stop refining an evidence group if the difference between each element of the cluster and the cluster average is below a chosen threshold. Details can be found in Algorithm 2.

Intuition suggests that if every evidence value is sufficiently close to its respective cluster mean, then the estimation error introduced by the coarse-to-fine approach should be relatively small when compared to LEPBP. In general, this requires some restrictions on the allowable potential functions if we aim to make this intuition precise.

Theorem 1. Fix $E > 0$ and $L > 0$. Let $E \subseteq V$ denote the set of continuous evidence variables with values given by v_E . If the evidence nodes are grouped into k sets S_1, \dots, S_k such that for all $a \in \{1, \dots, k\}$, $\max_i |v_i - \text{avg}(v_{S_a})| \leq E$ and $\forall c \in C$, $\log f_c$ is Lipschitz continuous with constant $L > 0$, then

$$\overline{Z_B}/Z_B \leq \exp(LE |c|),$$

where Z_B is the maximum BP estimate of the partition function using the approximate evidence and Z_B is the maximum estimate of the partition function for the given evidence.

Proof. (Sketch) Potential f_c depends on at most $|c|$ different evidence variables. Let \bar{v}_c correspond to the evidence values after averaging over each cluster. By Lipschitz continuity,

$$\begin{aligned} \log f_c(\bar{v}_c) &\leq \log f_c(v_c) + L|v_c - \bar{v}_c| \\ &\leq \log f_c(v_c) + LE |c|. \end{aligned}$$

Plugging this into the definition of the Bethe free energy [Yedidia *et al.*, 2001], whose maximum determines Z_B , yields the desired result. \square

Algorithm 2 Lifted EPBP with Coarse-to-Fine Lifting

```

1: Input: A factor graph  $G$  with variables  $X$ , factors  $f$ , evidence values  $v$ , and threshold  $E$ 
2: Return: The approximated belief  $b(x_i)$ 
3:  $S \subseteq 2^V \leftarrow$  initial clustering of evidence variables
4: Group variables with same domain and factors with same potential together
5: Run one iteration of CP to initialize  $G = (V, C)$ 
6: Initialize messages  $\hat{m}$  and proposals  $q$ 
7: repeat
8:   for each  $S_a \in S$  do
9:     if  $\max_{i \in S_a} |v_i - \text{avg}(v_{S_a})| > E$  then
10:      Run  $k_a$ -means with  $k=2$  to subdivide  $S_a$ 
11:   end if
12: end for
13: for  $i \in V$  do
14:   collect colors from neighbors of  $V_i$  and construct new groups according to color symmetries
15: end for
16: Update  $\hat{m}$  using one iteration of Algorithm 1 over  $G$ 
17: until convergence

```

Note that the log potential functions do not actually need to be Lipschitz over their entire domain for a similar argument to work; they only need to be locally Lipschitz over the evidence variables. This is a much weaker assumption, which is likely to be true in practice, e.g., Gaussian potentials.

4 Experimental Results

We aim to answer the following questions focusing on hybrid probabilistic relational models.

Q1: Does lifting significantly reduce the running time of particle based approaches for approximate inference?

Q2: Is LEPBP accurate when compared with alternative state-of-the-art methods for both exact and approximate inference?

Q3: Is LEPBP a practical tool for inference?

Q4: Does coarse-to-fine lifting improve upon LEPBP in practice?

Baselines. For comparison against LEPBP and Coarse-to-fine EPBP (C2FEPBP), we consider (1) the standard EPBP algorithm of Lienart *et al.* [2015], which is adapted here for generic inference in hybrid models, (2) the lifted Gaussian belief propagation algorithm of Ahmadi *et al.* [2011] as the message updates can be computed in closed form for this approximate inference method, and (3) the lifted relational Kalman filtering method of Choi *et al.* [2015] as it performs exact variable elimination in a lifted setting whenever all of the potential functions are Gaussian.

Implementation. EPBP, LEPBP, C2FEPBP, and LGaBP were implemented in Python 3.6, and all source code is available on GitHub¹. A MATLAB implementation of LRKF was obtained from Choi *et al.* [2015]. All experiments were performed on a machine with a 2.2 GHz Intel Core i7-8750H.

¹Code: github.com/leodd/Hybrid-Lifted-Belief-Propagation

	Algorithm	Time(s)	\mathcal{L}_1 Error
1% obs	EPBP	530.75	0.06 ± 0.01
	LEPBP	23.29	0.10 ± 0.02
	C2FEPBP	18.77	0.14 ± 0.04
5% obs	EPBP	509.52	0.06 ± 0.01
	LEPBP	113.25	0.09 ± 0.04
	C2FEPBP	64.94	0.08 ± 0.02
10% obs	EPBP	436.98	0.07 ± 0.01
	LEPBP	164.99	0.08 ± 0.02
	C2FEPBP	89.24	0.09 ± 0.01
20% obs	EPBP	381.92	0.05 ± 0.005
	LEPBP	261.26	0.04 ± 0.006
	C2FEPBP	133.14	0.08 ± 0.016

Table 1: Evaluation of lifted methods on relational Gaussian models with varying levels of evidence. Error was computed against LGaBP. The best performing methods are highlighted in bold.

CPU and 16 GB of memory. For sake of comparison, all algorithms, except LRKF, were restricted to a single core.

4.1 Relational Gaussian Models

We employ a continuous relational model with multivariate Gaussian potentials following the Recession model described by Choi *et al.* [2010]. The model has three bivariate Gaussian parfactors (i.e., parameteric factors) over two classes of instances (“bank” and “category”). We generate 100 “category” instances and 5 “bank” instances, which creates 606 grounded random variables. We randomly select 5%, 10%, and 20% of the variables in each instance as evidence and assign them evidence values sampled from the uniform distribution on $[-30, 30]$.

As the potentials are Gaussian, we used GaBP, which computes the BP message updates exactly, as a **strong** baseline for comparison. All message-passing algorithms were run for 15 iterations and sampling-based methods used 20 sampling points for the integral approximations. For coarse-to-fine lifting, we use k -means clustering with $k = 2$ for evidence group splitting and use dynamic splitting of the threshold which was initially being set to $E = \max_{S \in \mathcal{S}}(\text{avg}(v_{S_0}))$ and was decreased each iteration until $E = 0$. Each algorithm was run 5 times for each test, and the average CPU time and average \mathcal{L}_1 error compared to GaBP can be found in Table 1.

The results illustrate that, both LEPBP and C2FEPBP are significantly faster than EPBP on the original graph without increasing the average \mathcal{L}_1 error too much (EPBP draws more sampling points overall, which explains the accuracy difference with LEPBP and C2FEPBP). However, as the fraction of continuous evidence increases, the number of symmetries in the graph are significantly reduced, and the performance difference between the lifted methods and EPBP narrows. Still, the coarse-to-fine lifting method is roughly twice as fast as LEPBP as a result of reduced computation time in the early iterations when the evidence groupings are coarse. These observations suggest that **Q1**, at least when the amount of continuous evidence is relatively low, and **Q4** are true.

4.2 Relational Kalman Filtering

We next investigate approximate/exact inference methods for the Kalman filter, a widely applied tool, e.g., robotic control and object motion prediction, etc., to estimate the state of a dynamical system given evidence and controls. Specifically, given a set of state variables x , the Kalman filter represents the dynamic features of a state variable by $x_{t+1} = Ax_t + w$

$o_t = Cx_t + v$, where A is a transition matrix, C is an observation matrix, $w \sim \mathcal{N}(0, Q)$, and $v \sim \mathcal{N}(0, R)$. The covariance matrices Q and R relate to the transitions and observations respectively. We can represent the univariate transition model of a Kalman Filter with the linear Gaussian $\exp((x_{t+1} - Ax_t)^2 / \sigma^2)$. For the multivariate transition model, we decompose the linear Gaussian potential into several pairwise potentials. The relational Kalman filter, [Choi *et al.*, 2011; Choi *et al.*, 2015], represents the transition model in a lifted (parameterized) way, where groups of similar state (correspondingly observation) variables share the same parameters for the transitions (correspondingly observations), *also known as parameter tying*.

We used groundwater level data extracted from the Republican River Compact Association model [McKusick, 2003], which is a monthly record of the measured head position of 3420 wells over 850 months. As in Choi *et al.* [2015], we grouped wells in same area together and assume that wells in the same group share the same transition and observation model. We tested two cases 1) a tree-structured model 2) a cyclic model. For the first case, we select one group of wells that has 76 instances and defined a model – $A = \alpha I$, $C = I$, $Q = \beta I$, $R = \gamma I$ where $\alpha \sim U(0.5, 1)$, $\beta \sim U(5, 10)$, $\gamma \sim U(1, 5)$ and I is the identity matrix. For the second task, we constructed two groups of wells with 3 instances each. We define model similarly, except that $A = \alpha I + 0.01$ and $Q = \beta J$, where J is the matrix of ones. We construct a Kalman filter model with 20 time steps and use 20 months of data as the observations. We evaluated the message-passing algorithms with 20 iterations and the sampling-based algorithms used 50 particles to approximate the integrals. For evidence splitting threshold E , we used the same setting as in Section 4.1.

We compared the MAP value of variables in the last time step. As can be seen from Table 2, the difference between the result of our approaches and the result of exact LRKF and LGaBP are significantly smaller. The range of the continuous values are between [-5,5] for the tree case and between [-1,9] for the loopy case. Our methods have a less than 1% error in the tree case and around 3% error in the loopy case even without making any restrictive assumptions about the potentials, thus answering **Q2** affirmatively.

4.3 Hybrid Markov Logic Networks

To demonstrate the efficiency of our approach in truly hybrid models, following the hybrid MLN work of Wang and Domingos [2008], we simulated an *advisedBy* domain where we employed two clauses: $1.0 : \text{author}(P, Pa) \wedge \text{firstAuthor}(S, Pa) \rightarrow \text{advisedBy}(S, P)$ and $0.1 : \text{advised}(S, P) [\text{minSchool}(S) = \text{pLength}(P)]$. The first clause states that a professor advises a student if they are coauthors in a paper where the student is a primary author.

	Algorithm	\mathcal{E}_1 Diff.(LRKF)	\mathcal{E}_1 Diff.(LGaBP)
Tree	EPBP	0.12 ± 0.05	0.12 ± 0.06
	LEPBP	0.12 ± 0.10	0.11 ± 0.10
	C2FEPBP	0.09 ± 0.09	0.10 ± 0.08
Cycle	EPBP	0.31 ± 0.03	0.27 ± 0.07
	LEPBP	0.32 ± 0.04	0.26 ± 0.08
	C2FEPBP	0.34 ± 0.06	0.29 ± 0.06

Table 2: Accuracy of lifted Particle methods on relational Kalman filter models against LRKF (exact) and LGaBP.

Algorithm	Time(s)	\mathcal{E}_1 Diff.
EPBP	15.57	-
LEPBP	7.9	0.35 ± 0.28
C2FEPBP	7.17	0.36 ± 0.25

Table 3: Performance on the hybrid MLN where the query is on the months in school and the project length. Rough accuracy was computed against the average value of 5 EPBP runs.

This is a hybrid MLN as the second clause is a soft equality, i.e., $\alpha = \beta$ is interpreted as a shorthand for $(\alpha \beta)^2$. Essentially this is a Gaussian penalty with the the mean being the equality constraint of months in school (*minSchool*) of the student and project length *pLength*, i.e., $-(\text{minSchool} \cdot \text{pLength})^2$ and a variance of 0.1. Note that since there does not exist a hybrid lifted inference method to compare against, we employ the average of five runs of EPBP as the baseline and compute the difference to EPBP by our methods.

We generated 100 students, 20 Professors and 50 papers for testing. We sampled 20% of the students from a uniform distribution and set evidence *e* value *minSchool*(*S*), where $e \sim \mathcal{N}(5)$. The assignments of authorships were performed based on sampling of the first rule. The queries include the months in school for the other 80 students and the project lengths of the Professors. The message-passing algorithms were run for 10 iterations and sampling-based methods used 20 particles. For evidence splitting threshold *E*, we used the same setting as in Section 4.1. As can be seen from Table 3, the proposed lifted methods are significantly faster than EPBP even on this smaller network while achieving nearly the same performance (the range of values is 30 and hence, the methods have around 1% difference to EPBP). This allows us to answer **Q3** affirmatively in that our approach appears to be promising for real tasks where hybrid SRL models need to be employed.

4.4 Image Denoising

As a final task, we compared the running times of EPBP, LEPBP, and coarse-to-fine LEPBP on a simple image denoising task [Lienart *et al.*, 2015], which is challenging for regular lifting as the evidence essentially breaks all model symmetries. For this task, a 50×50 grayscale image has been corrupted with Gaussian noise and the aim is to reconstruct the original image. We construct a grid MRF over the image with a random variable for each pixel whose real value is assumed to be hidden while the noisy image pixel itself is treated as the observation. Following Lienart *et al.* [2015], we define a potential between hidden vari-

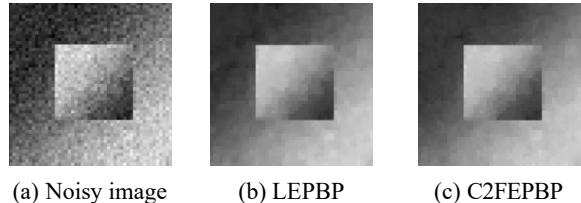


Figure 1: Denoising of the 50×50 image (a).

ables $f(x_i, x_j) \propto \exp(-\alpha \max(|x_i - x_j|, \beta))$ and potential between hidden variable and observation node $f(x_i, o_i) \propto \exp(-(x_i - o_i)^2 / 2\sigma^2)$, where we choose parameters $\alpha = 0.285$, $\beta = 25$ and $\sigma = 2.24$.

In this example, we tested EPBP, LEPBP, and coarse-to-fine lifting, running all three algorithms for 10 iterations. The sampling methods used 10 particles and coarse-to-fine lifting used the same splitting threshold *E* as in Section 4.1. Figure 1 shows the resulting denoisings. The CPU time for all three algorithms are: (EPBP) 109.5312s, (LEPBP) 117.9062s, (C2FEPBP) 60.6562s. The result for EPBP was not shown since, in this instance, LEPBP are the same as EPBP as regular lifting will return the grounded graph. The denoised result of C2FEPBP is very close to the one of LEPBP, but with almost half CPU time of LEPBP. This not only answers **Q4** affirmatively, in that the lifting improves upon LEPBP, but also allows for the algorithm to be potentially employed in practical tasks (**Q3**).

5 Conclusion

We presented the first lifted inference method for hybrid relational probabilistic models. Our proposed approach consists of two key steps, constructing the compressed network and then running a modified particle BP method. The key aspect of our work is that we do not make any restrictive assumptions on the potentials (the only assumption is that the integrals exist). We demonstrated the effectiveness both theoretically (under locality assumptions) and empirically in several different settings. Although all of the sampling methods are slower than GaBP on a single core, they have the flexibility to be run over generic potential functions and can be efficiently parallelized, an important future direction. We are currently exploring variational methods, similar in spirit to Guo *et al.* [2019], to scale lifted inference to large hybrid domains as well as POMDPs, where such algorithms can be used for efficient tracking and uncertainty modeling.

Acknowledgments

This work was supported, in part, by the DARPA Explainable Artificial Intelligence (XAI) program (N66001-17-2-4032) and NSF grant III-1527312. SN acknowledges the support of NSF grant IIS-1836565 and AFOSR award FA9550-18-1-0462. Any opinions, findings and conclusion or recommendations are those of the authors and do not necessarily reflect the view of the DARPA, AFOSR or the US government.

References

[Ahmadi *et al.*, 2011] Babak Ahmadi, Kristian Kersting, and Scott Sanner. Multi-evidence lifted message passing, with application to PageRank and the Kalman filter. In *IJCAI*, 2011.

[Berrou *et al.*, 1993] Claude Berrou, Alain Glavieux, and Punya Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. In *ICC*, 1993.

[Choi *et al.*, 2010] Jaesik Choi, Eyal Amir, and David J. Hill. Lifted inference for relational continuous models. In *UAI*, 2010.

[Choi *et al.*, 2011] Jaesik Choi, Abner Guzman-Rivera, and Eyal Amir. Lifted relational Kalman filtering. In *AAAI*, 2011.

[Choi *et al.*, 2015] Jaesik Choi, Eyal Amir, Tianfang Xu, and Albert J. Valocchi. Learning relational Kalman filtering. In *AAAI*, 2015.

[de Salvo Braz *et al.*, 2005] Rodrigo de Salvo Braz, Eyal Amir, and Dan Roth. Lifted first-order probabilistic inference. In *IJCAI*, 2005.

[den Broeck, 2013] Guy Van den Broeck. *Lifted Inference and Learning in Statistical Relational Models*. PhD thesis, 2013.

[Getoor and Taskar, 2007] Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[Gogate and Domingos, 2011] Vibhav Gogate and Pedro Domingos. Probabilistic theorem proving. In *UAI*, 2011.

[Guo *et al.*, 2019] Yuanzhen Guo, Hao Xiong, and Nicholas Ruozzi. Marginal inference in continuous Markov random fields using mixtures. In *AAAI*, 2019.

[Habeeb *et al.*, 2017] Haroun Habeeb, Ankit Anand, Mausam, and Parag Singla. Coarse-to-fine lifted MAP inference in computer vision. In *IJCAI*, 2017.

[Ihler and McAllester, 2009] Alexander Ihler and David McAllester. Particle belief propagation. In *AISTATS*, 2009.

[Kersting *et al.*, 2009] Kristian Kersting, Babak Ahmadi, and Sriraam Natarajan. Counting belief propagation. In *UAI*, 2009.

[Koller, 1999] Daphne Koller. Probabilistic relational models. In *ILP*, 1999.

[Lienart *et al.*, 2015] Thibaut Lienart, Yee Whye Teh, and Arnaud Doucet. Expectation particle belief propagation. In *NeurIPS*, 2015.

[Malioutov *et al.*, 2006] Dmitry M. Malioutov, Jason K. Johnson, and Alan S. Willsky. Walk-sums and belief propagation in Gaussian graphical models. *Journal of Machine Learning Research*, 7:2031–2064, 2006.

[McKusick, 2003] Vincent L. McKusick. Final report for the special master with certificate of adoption of RRCA groundwater model. 2003.

[Minka, 2001] Thomas P. Minka. Expectation propagation for approximate Bayesian inference. In *UAI*, 2001.

[Moallemi and Roy, 2009] Ciamac C. Moallemi and Benjamin Van Roy. Convergence of min-sum message passing for quadratic optimization. *Information Theory, IEEE Transactions on*, 55(5):2413–2423, May 2009.

[Murphy *et al.*, 1999] Kevin Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, 1999.

[Niepert, 2012] Mathias Niepert. Lifted probabilistic inference: An mcmc perspective. In *UAI*, 2012.

[Poole, 2003] David Poole. First-order probabilistic inference. In *IJCAI*, 2003.

[Raedt *et al.*, 2016] Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*. 2016.

[Richardson and Domingos, 2006] Matthew Richardson and Pedro Domingos. Markov logic networks. 62, 2006.

[Ruozzi and Tatikonda, 2013] Nicholas Ruozzi and Sekhar Tatikonda. Message-passing algorithms for quadratic minimization. *Journal of Machine Learning Research (JMLR)*, 14:2287–2314, 2013.

[Sarkhel *et al.*, 2015] Somdeb Sarkhel, Parag Singla, and Vibhav G. Gogate. Fast lifted map inference via partitioning. In *NeurIPS*, 2015.

[Singla and Domingos, 2008] Parag Singla and Pedro Domingos. Lifted first-order belief propagation. In *AAAI*, 2008.

[Sudderth *et al.*, 2010] Erik B. Sudderth, Alexander T. Ihler, William T. Freeman, and Alan S. Willsky. Nonparametric belief propagation. *Communications of the ACM*, 53(10), Oct. 2010.

[Venugopal and Gogate, 2012] Deepak Venugopal and Vibhav Gogate. On lifting the gibbs sampling algorithm. In *NIPS*, 2012.

[Venugopal and Gogate, 2014] Deepak Venugopal and Vibhav Gogate. Evidence-based clustering for scalable inference in markov logic. In *ECML PKDD*, 2014.

[Wang and Domingos, 2008] Jue Wang and Pedro Domingos. Hybrid markov logic networks. In *AAAI*, 2008.

[Wang and Song, 2011] Haizhou Wang and Mingzhou Song. Ckmeans. 1d. dp: optimal k-means clustering in one dimension by dynamic programming. *The R journal*, 3(2), 2011.

[Weiss and Freeman, 2000] Yair Weiss and William T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. In *NeurIPS*, 2000.

[Yedidia *et al.*, 2001] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Bethe free energy, Kikuchi approximations, and belief propagation algorithms. *NeurIPS*, 2001.