Gauged Mini-Bucket Elimination for Approximate Inference

Sungsoo Ahn
Korea Advanced Institute
of Science and Technology

Michael Chertkov
Los Alamos
National Laboratory,
Skolkovo Institute of
Science and Technology

Jinwoo Shin
Korea Advanced Institute
of Science and Technology

Adrian Weller
University of Cambridge,
The Alan Turing Institute

Abstract

Computing the partition function Z of a discrete graphical model is a fundamental inference challenge. Since this is computationally intractable, variational approximations are often used in practice. Recently, so-called gauge transformations were used to improve variational lower bounds on Z. In this paper, we propose a new gauge-variational approach, termed WMBE-G, which combines gauge transformations with the weighted mini-bucket elimination (WMBE) method. WMBE-G can provide both upper and lower bounds on Z, and is easier to optimize than the prior gauge-variational algorithm. We show that WMBE-G strictly improves the earlier WMBE approximation for symmetric models including Ising models with no magnetic field. Our experimental results demonstrate the effectiveness of WMBE-G even for generic, non-symmetric models.

1 INTRODUCTION

Graphical Models (GMs) express the factorization of the joint multivariate probability distribution over subsets of variables via graphical relations among them. GMs have been developed in information theory [1, 2], physics [3, 4, 5, 6, 7], artificial intelligence [8], and machine learning [9, 10]. For a GM, computing the partition function Z (the normalization constant) is a fundamental inference task of great interest. However, this task is known to be computationally intractable in general: it is #P-hard even to approximate [11].

Variational approaches frame the inference task as an optimization problem, which is typically solved approximately. Key challenges for variational methods are to scale efficiently with the number of variables; and to try to provide guaranteed upper or lower bounds on Z.

Popular variational methods include: the mean-field (MF) approximation [6], which provides a lower bound on Z; the tree-reweighted (TRW) approximation [12], which provides an upper bound; and belief propagation (BP) [13], which often performs well but provides neither an upper nor lower bound in general. Other variational methods have been investigated for providing lower bounds [14, 15, 16, 17] or upper bounds [15, 16, 17] for approximating Z.

Methods using *reparametrizations* [18], *gauge transformations* (GT) [19, 20] or *holographic transformations* (HT) [21, 22] have been explored. These methods each consider modifying the base GM by transforming the potential factors in various ways, aiming to simplify the inference task, while keeping the partition function Z unchanged. We call these methods collectively *Z-invariant methods*. See [23, 24, 25] for discussions of the differences and relations between these methods.

An approach to combine variational and Z-invariant methods was recently introduced by [26], yielding a lower bound on Z. They proposed gauge-variational optimization formulations built upon MF and BP, incorporating the generic IPOPT solver [27] as an essential inner optimization routine. Here we introduce a new gauge-variational optimization approach, using variational methods other than MF and BP, and employing a specialized solver for inner optimization which is more efficient than IPOPT. Further, our approach yields lower and upper bounds on Z.

Contribution. We develop a new family of gauge-variational algorithms combining the methods of gauge transformations (GTs) and weighted mini-bucket elimination (WMBE) [16]. The significance of our new approach, which we call WMBE-G, is twofold:

$\mathcal{C}1$. We introduce optimization formulations which provide both upper and lower bounds of Z by general-

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

izing the original WMBE bounds to incorporate GTs. The authors [16] use the re-parameterization framework, which is a distribution-invariant method that is a strict sub-class of GTs. Hence, our formulations explore a strictly larger freedom in optimization, which we observe typically leads to significantly better bounds in practice. Indeed, we provide an analytic class of GMs (symmetric binary GMs including Ising models with no magnetic field) where ours provide strictly better results.

$\mathcal{C}2$. We propose a novel optimization solver alternating between gauges and factors to minimize (or maximize) the proposed objectives, and demonstrate its computational advantages. We remark that the earlier optimization approaches in [26] required 'non-negativity' constraints which are tricky to handle, while we do not. [26] addresses the challenge using the generic IPOPT solver with the log-barrier method, but it is not clear if this will scale well for large instances. On the other hand, our proposed algorithms are clearly scalable since they solve purely unconstrained optimizations in a distributed manner.

Our experimental results show that WMBE-G has superior performance in comparison with other known algorithms, including WMBE. We remark that the main contribution of WMBE [16] was to introduce Hölder weights to improve the original mini-bucket elimination (BE) bound [28], whereas we additionally optimize gauges for even better performance. In our experiments, we observe that the contribution of Hölder weights is relatively marginal compared to gauges in optimizing the BE bound (see Section 4 for more details). Namely, we found that gauges are more crucial than Hölder weights for better approximation to Z, while the computational costs of optimizing them are similar. In this paper, we mainly focus on WMBE-G using the Hölder inequality to obtain an upper bound on Z, but a lower bound can be similarly derived using the reverse Hölder inequality (see Section 2.3).

2 PRELIMINARIES

2.1 Graphical Models

Factor-graph GM. We consider an undirected, bipartite factor graph $G = (V, E)$ with vertices $V = X \cup F$ comprising variables X and factors F, and edges between variable and factor nodes $E \subseteq X \times F$. Each random variable $x_v \in X$ is discrete, taking values in $\{1, \cdots, d\}$. The distribution factorizes as follows:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{\alpha \in F} f_\alpha(\mathbf{x}_\alpha). \tag{1}$$

Here, $\mathcal{F} = \{f_\alpha\}_{\alpha \in F}$ is a set of non-negative functions called *factors*, and $\mathbf{x}_\alpha$ is the subset of variables for factor α, i.e., $\mathbf{x}_\alpha = [x_v : v \in N(\alpha)]$ with $N(\alpha) = \{v : (v, \alpha) \in E\}$. The normalization constant

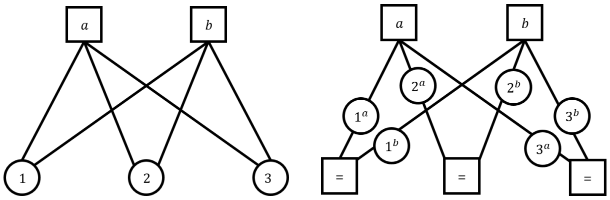


Figure 1: Example of transformation from the factor-graph GM (left) to the Forney-style GM (right). Squares and circles indicate factors and variables respectively. New factors denoted as '=' force adjoining variables be consistent, i.e., have the same value.

$$Z := \sum_{\mathbf{x}} \prod_{\alpha \in F} f_\alpha(\mathbf{x}_\alpha)$$

is called the *partition function*. It is well known that the partition function is computationally intractable in general: it is #P-hard even to approximate [11].

Forney-style GM. For ease of notation in dealing with GTs, throughout this paper we shall assume Forney-style GMs [29]. These ensure that every variable has two adjacent factors, i.e., $|N(v)| = 2 \ \forall \ v \in X$. As shown in [19, 20], Forney-style GMs provide a more compact description of gauge transformations without any loss of generality: given any factor-graph GM, one can construct an equivalent Forney-style GM [30]. See Figure 1 for an example.

2.2 Gauge Transformations

Gauge transformations [19, 20] are a family of linear transformations of the factor functions in (1) which leave the the partition function Z invariant. GTs are defined by the following set of invertible $d \times d$ matrices $\{G_{v\alpha} : (v, \alpha) \in E\}$, termed *gauges*:

$$G_{v\alpha} = \begin{bmatrix} G_{v\alpha}(1,1) & \cdots & G_{v\alpha}(1,d) \\ \vdots & \ddots & \vdots \\ G_{v\alpha}(d,1) & \cdots & G_{v\alpha}(d,d) \end{bmatrix}.$$

The transformed GM with respect to the gauges $\mathbf{G} = \{G_{v\alpha} : (v, \alpha) \in E\}$ consists of modified factors $\{\widehat{f}_\alpha : \alpha \in F\}$ computed as follows:

$$\widehat{f}_\alpha(\mathbf{x}_\alpha; \mathbf{G}_\alpha) = \sum_{\mathbf{x}'_\alpha} f_\alpha(\mathbf{x}'_\alpha) \prod_{v \in N(\alpha)} G_{v\alpha}(x_v, x'_v), \tag{2}$$

where $\mathbf{G}_\alpha = \{G_{v\alpha} : v \in N(\alpha)\}$. Here, the gauges must satisfy the following *gauge constraints*:

$$G_{v\alpha}^\top G_{v\beta} = \mathbb{I}, \qquad \forall v \in X, \tag{3}$$

where $\mathbb{I}$ is the identity matrix and $N(v) = \{\alpha, \beta\}$ (recall that we assume $|N(v)| = 2$). With these constraints, the partition function is known to be invariant under the transformation [19, 20], i.e.,

$$Z = \sum_{\mathbf{x}} \prod_{\alpha \in F} f_\alpha(\mathbf{x}_\alpha) = \sum_{\mathbf{x}} \prod_{\alpha \in F} \widehat{f}_\alpha(\mathbf{x}_\alpha; \mathbf{G}_\alpha).$$

Thus gauges lead to the transformed distribution $p(\mathbf{x}; \mathbf{G}) = \prod_{\alpha \in F} \widehat{f}_\alpha(\mathbf{x}_\alpha; \mathbf{G}_\alpha)/Z$. We remark that it might be invalid when $\widehat{f}_\alpha(\mathbf{x}_\alpha; \mathbf{G}_\alpha)$ is negative. Nevertheless, even in this case, the partition function invariance still holds. We provide an example of a gauge transformation in the Supplement.

2.3 Weighted Mini-Bucket Elimination

Bucket (or variable) elimination (BE) [31, 32] is a method for computing the partition function exactly based on directly summing out the variables sequentially. First, BE assumes a fixed elimination ordering $o = [v_1, \cdots, v_n]$ among variables nodes $v \in X$. Then BE groups factors by placing each factor f_α in the "bucket" B_v of its earliest argument $v \in N(\alpha)$ appearing in the elimination order o. Next, BE eliminates the variable by marginalizing the product of factors in the bucket, i.e.,

$$f_{B_v}(\mathbf{x}_{B_v}) = \sum_{x_v} \prod_{f_\alpha \in B_v} f_\alpha(\mathbf{x}_\alpha) \qquad \forall\, \mathbf{x}_{B_v}, \tag{4}$$

where $\mathbf{x}_{B_v} = [x_u : u \in var(B_v), u \neq v]$ and $var(B_v)$ indicates the subset of variables in the bucket. Finally, the newly generated function f_{B_v} is inserted into another bucket corresponding to its earliest argument in the elimination order. This process is easily seen as applying a distributive property: groups of factors corresponding to buckets are summed out sequentially, and then the newly created factor (without the eliminated variable) is assigned to another bucket.

The computational cost of BE is exponential in the number of uneliminated variables in the bucket, i.e., the *induced width*[1] of the graph given the elimination order. BE is summarized in Algorithm 1.

Mini-bucket elimination (MBE) [28] and *weighted mini-bucket elimination* (WMBE) [16] approximate BE by splitting computation of each bucket into several "mini-buckets", where WMBE additionally makes use of Hölder's inequality [33]. Since MBE is a special case of WMBE (by choosing extreme Hölder weights), here we focus on providing background for WMBE.

Algorithm 1 BE for computing Z

```
Input: GM on graph $G = (V, E)$ with $V = (X, F)$ and factors $\mathcal{F} = \{f_\alpha\}_{\alpha \in F}$ and elimination order $o = [v_1, \cdots, v_n]$.
$\mathcal{F}' \leftarrow \mathcal{F}$
for $v$ in $o$ do
    $B_v \leftarrow \{f_\alpha | f_\alpha \in \mathcal{F}, v \in N(\alpha)\}$
    Generate new factor by:
        $f_{B_v}(\mathbf{x}_{B_v}) = \sum_{x_v} \prod_{f_\alpha \in B_v} f_\alpha(\mathbf{x}_\alpha), \ \forall\, \mathbf{x}_{B_v}.$
    $\mathcal{F}' \leftarrow \mathcal{F}' \cup \{f_{B_v}\} - B_v$
end for
Output: $Z = \prod_{f_\alpha \in \mathcal{F}'} f_\alpha$
```

Let $\{\psi_i(x), i = 1, \cdots, m\}$ be some functions defined on discrete variable x, and $\mathbf{w} = [w_1, \cdots w_n]$ be a vector of *Hölder weights*. We define a *weighted absolute summation*, defined as follows:

$$\sum_{x}^{w_i} \psi_i(x) := \Big(\sum_x |\psi_i(x)|^{1/w_i}\Big)^{w_i}.$$

Equivalently, $\sum_x^{w_i}$ is the Schatten p-norm with $p = 1/w_i$. If $w_i > 0$ for all $i \geq 1$, then Hölder's inequality implies that

$$\sum_{x}^{w_0} \prod_{i=1}^{m} \psi_i(x) \leq \prod_{i=1}^{m} \sum_{x}^{w_i} \psi_i(x), \tag{5}$$

where $w_0 = \sum_i w_i$. If only one weight is positive, e.g., $w_1 > 0$ and $w_i < 0$ for all $i > 1$, we have the reverse Hölder's inequality:

$$\sum_{x}^{w_0} \prod_{i=1}^{m} \psi_i(x) \geq \prod_{i=1}^{m} \sum_{x}^{w_i} \psi_i(x). \tag{6}$$

WMBE modifies BE by applying Hölder's inequality whenever the size of a bucket, i.e., length of $\mathbf{x}_{B_v}$, exceeds some given parameter called *ibound*. In this case, WMBE splits the bucket into multiple 'mini-buckets', and weighted absolute summation is evaluated sequentially in place of (4), i.e.,

$$\sum_{x_v} \prod_{f_\alpha \in B} |f_\alpha(\mathbf{x}_\alpha)| \leq \prod_{r=1}^{R_v} \sum_{x_v}^{w_r} \prod_{f_\alpha \in B_v^r} f_\alpha(\mathbf{x}_\alpha),$$

for all $\mathbf{x}_{B_v}$, where Hölder weights satisfy $\sum_r w_r = 1, w_r > 0$, $B_v = \bigcup_r B_v^r$, and B_v^r is disjoint for all r. We then generate multiple new factors by:

$$f_{B_v^r}(\mathbf{x}_{B_v^r}) = \sum_{x_v}^{w_r} \prod_{f_\alpha \in B_v^r} f_\alpha(\mathbf{x}_\alpha), \qquad \forall\, \mathbf{x}_{B_v},$$

and insert into other buckets. By construction, WMBE yields an upper bound for the partition function Z. One

[1] The minimum possible induced with is called *tree-width*.

Algorithm 2 WMBE for bounding Z

```
Input: GM on graph $G = (V, E)$ with $V = (X, F)$, factors $\mathcal{F} = \{f_\alpha\}_{\alpha \in F}$, elimination order $o = [v_1, \cdots, v_n]$ and bound on bucket size $ibound$.
$\mathcal{F}' \leftarrow \mathcal{F}$
for $v$ in $o$ do
    $B_v \leftarrow \{f_\alpha | f_\alpha \in \mathcal{F}', v \in \partial\alpha\}$
    Partition $B_v$ into $R_v$ subgroups $\{B_v^r\}_{r=1}^{R_v}$ such that $|var(B_v^r)| \le ibound$ for all $r$.
    Assign weights $w_1, \cdots, w_{R_v}$ while satisfying $\sum_r w_r = 1$.
    for $r \leftarrow 1, \cdots, R_v$ do
        Generate a new factor by:
            $$f_{B_v^r}(\mathbf{x}_{B_v^r}) = \sum_{x_v}^{w_r} \prod_{f_\alpha \in B_v^r} f_\alpha(\mathbf{x}_\alpha), \; \forall\, \mathbf{x}_{B_v}.$$
        $\mathcal{F}' \leftarrow \mathcal{F}' \cup \{f_{B_v^r}\} - B_v^r$
    end for
end for
Output: $Z_{\text{WMBE}} = \prod_{f_\alpha \in \mathcal{F}'} f_\alpha$
```

can use the same idea to derive a lower bound for Z using the reverse Hölder's inequality. We summarize WMBE in Algorithm 2.

One can interpret MBE as a special case of WMBE by assigning a single weight to be close to 1 and others to be close to 0, i.e., $\mathbf{w} = \lim_{w \to 0^+} [1 - w, w, w, \cdots]$. Instead, Liu and Ihler [16] optimize the Hölder weights so that WMBE can outperform MBE, which we discuss further in Section 3.

3 GAUGED WMBE ALGORITHM

In this section, we describe our gauge optimization scheme WMBE-G to improve the previous WMBE bound, yielding gauranteed upper bound approximations for the partition function Z. Our scheme improves the standard WMBE bound by searching over the large family of gauge transformed (possibly invalid) GMs to find the tightest WMBE bound possible.

3.1 Key Optimization Formulation

In order to describe the optimization formulation for tightening the WMBE bound, we first observe that (8) can be reformulated into

$$\sum_{x_v} \prod_{f_\alpha \in B} |f_\alpha(\mathbf{x}_\alpha)| \le \sum_{x_v^{(1)}}^{w_1} \cdots \sum_{x_v^{(R_v)}}^{w_{R_v}} \prod_{r=1}^{R_v} \prod_{\alpha \in B_v^r} f_\alpha(\mathbf{x}_{\alpha \setminus v}, x_v^{(r)}) \tag{7}$$

where $\mathbf{x}_{\alpha\setminus v} = [x_u : u \in N(\alpha), u \neq v]$. While notation is complex, this is simply applying the distributive property on the right hand side of (8). The procedure can be seen as 'splitting' variable from x_v to $x_v^{(1)}, \cdots x_v^{(R_v)}$ and its associated node from v to $v^{(1)}, \cdots v^{(R_v)}$ so that factors no longer share the split variable. We remark that under Forney-style GMs, $R_v \le 2$ since exactly 2 factors are associated with a variable. After repeatedly applying the inequality, we arrive at the following WMBE bound, termed *weighted partition function*:

$$Z \le Z_{\text{WMBE}} = \sum_{\bar{x}_{\bar{1}}}^{\bar{w}_{\bar{n}}} \cdots \sum_{\bar{x}_1}^{\bar{w}_1} \prod_{\alpha \in F} f_\alpha(\bar{\mathbf{x}}_\alpha). \tag{8}$$

In (8), $\bar{\mathbf{x}} = [\bar{x}_1, \cdots, \bar{x}_{\bar{n}}]$ and $\bar{\mathbf{w}} = [\bar{w}_1, \cdots, \bar{x}_{\bar{n}}]$ indicate the 'split' version of variables and associated Hölder weights, indexed by appearance of associated node in the modified elimination order $\bar{o} = [v_1^{(1)}, \cdots v_1^{(R_{v_1})}, \cdots, v_n^{(1)}, \cdots v_n^{(R_{v_n})}]$. Therefore, the WMBE bound can be seen as a weighted absolute summation over product of factors in a new GM. However, unlike the original partition function, the weighted absolute summation is tractable with respect to $ibound$ since at most d^{ibound} terms are counted for each weighted absolute summation, or equivalently variable elimination of mini-buckets. Finally, we are able to present our main optimization formulation:

$$\begin{aligned} \underset{\mathbf{G}}{\text{minimize}} \quad & \sum_{\bar{x}_{\bar{n}}}^{\bar{w}_{\bar{n}}} \cdots \sum_{\bar{x}_1}^{\bar{w}_1} \prod_{\alpha \in F} \widehat{f}_\alpha(\bar{\mathbf{x}}_\alpha; \mathbf{G}_\alpha), \\ \text{subject to} \quad & G_{v\alpha}^\top G_{v\beta} = \mathbb{I}, \quad \forall\, v \in X, N(v) = \{\alpha, \beta\}. \end{aligned} \tag{9}$$

3.2 Algorithm Description

We now describe an efficient algorithm to optimize (9). First, the gauge constraint can be removed simply by expressing one (of the two) gauges in terms of the other, e.g., $G_{v\beta}$ via $(G_{v\alpha}^\top)^{-1}$. Then, (9) can be optimized via any type of unconstrained optimization solver. Here, we optimize gauges by gradient descent followed by additional updates on factor values.

To this end, we initialize gauges by identity matrices, which immediately yields the original WMBE bound from (8) since $f_\alpha(\mathbf{x}_\alpha) = \widehat{f}_\alpha(\mathbf{x}_\alpha; \mathbb{I}_\alpha)$, where $\mathbb{I}_\alpha = [G_{v\alpha} = \mathbb{I} : (v, \alpha) \in E]$. Next, under expressing gauges via one another, i.e., $G_{v\beta} \leftarrow \left(G_{v\alpha}^\top\right)^{-1}$, we update each gauge element by gradient descent for minimization of the weighted log partition function upper bound $\log Z_{\text{WMBE}}$ as follows:

$$\begin{aligned} G_{v\alpha}(x'_v, x''_v) &\leftarrow G_{v\alpha}(x'_v, x''_v) - \mu \frac{\partial \log Z_{\text{WMBE}}}{\partial G_{v\alpha}(x'_v, x''_v)} \\ \frac{\partial \log Z_{\text{WMBE}}}{\partial G_{v\alpha}(x'_v, x''_v)} &= \sum_{\bar{\mathbf{x}}_{\alpha\setminus v}} q(\bar{\mathbf{x}}_{\alpha\setminus v}, x''_v) \frac{f_\alpha(\bar{\mathbf{x}}_{\alpha\setminus v}, x'_v)}{f_\alpha(\bar{\mathbf{x}}_{\alpha\setminus v}, x''_v)} \\ &\quad - \sum_{\bar{\mathbf{x}}_{\beta\setminus v}} q(\bar{\mathbf{x}}_{\beta\setminus v}, x'_v) \frac{f_\beta(\bar{\mathbf{x}}_{\beta\setminus v}, x''_v)}{f_\beta(\bar{\mathbf{x}}_{\beta\setminus v}, x'_v)}, \end{aligned} \tag{10}$$

where μ is the step size[2], $\mathbf{x}_{\alpha\setminus v} = [x_u : u \in N(\alpha), u \neq v]$ and q is an 'auxiliary distribution' defined as

$$q(\bar{\mathbf{x}}) = \prod_{k=1}^{\bar{n}} q(\bar{x}_k | \bar{x}_{k+1:\bar{n}}),$$

$$q(\bar{x}_k | \bar{x}_{k+1:n}) \propto \left(\sum^{\bar{w}_{k-1}}_{\bar{x}_{k-1}} \cdots \sum^{\bar{w}_1}_{\bar{x}_1} \prod_{\alpha \in F} f_\alpha(\bar{\mathbf{x}}_\alpha) \right)^{1/\bar{w}_k}.$$

We also update $G_{v\beta} \leftarrow (G_{v\alpha}^\top)^{-1}$ and the value of associated factors by the gauge-transformed factors, i.e.,

$$f_\alpha(\mathbf{x}_\alpha) \leftarrow \widehat{f}_\alpha(\mathbf{x}_\alpha; \mathbf{G}_\alpha), \tag{11}$$

and similarly for f_β. Finally, for the next iteration, we reset $G_{v\alpha} \leftarrow \mathbb{I}$.

The above update leads to an improved WMBE bound, which can be repeated for better bounds (until convergence). Each iteration $t = 1, \ldots T$ results in a sequence of gauges $\mathbf{G}^{(t)}$ obtained by (10), and factors $f_\alpha^{(t)}$ obtained by (11) can be expressed as $f_\alpha^{(t)}(\mathbf{x}_\alpha) = \widehat{f}_\alpha^{(0)}(\mathbf{x}_\alpha; \mathbf{G}'_\alpha)$, where $f_\alpha^{(0)} = f_\alpha$ is the original GM factor, and $\mathbf{G}'_\alpha$ consists of gauges $G'_{v\alpha} = G_{v\alpha}^{(t+1)} G_{v\alpha}^{(t)} \cdots G_{v\alpha}^{(1)}$ for $v \in N(\alpha)$. We remark that one can use naïve gradient descent, i.e., update gauges only (without resetting to identity matrices), instead of factors as in (11). However, by utilizing the additional factor updates, the gradient formulation is simplified and redundant computations of gauge transformations are reduced. We summarize the above update procedure in Algorithm 3.

Furthermore, one can utilize ideas from [16] in order to improve the efficiency and power of the proposed optimization. First, computation of auxiliary marginals $q(\mathbf{x}_\alpha)$ in (10) can be efficiently carried out by a message-passing scheme proposed by the authors. Moreover, one can jointly optimize the Hölder weights $\bar{\mathbf{w}}$ in addition to $\mathbf{G}$ using the auxiliary distribution during optimization of (9). In our experiments, we utilize both the message-passing algorithm and the joint optimization involving $\bar{\mathbf{w}}$ using the log-gradient step proposed by the authors.

Finally, we remark that the elimination order and bucket split strategy might be another freedom that one may exploit in order to tighten the WMBE bound. However, their optimizations are hard (see [16]). Hence, we choose the elimination order arbitrarily in our experiments. For the bucket split strategy, if one assumes Forney-style GMs, any strategy reduces into a fixed split process, i.e., whenever *ibound* is exceeded, a variable x is always split in two parts $x^{(1)}, x^{(2)}$, and adjacent factors are assigned separately.

Algorithm 3 Gauged WMBE for bounding Z

```
Input: GM on graph $G = (V, E)$ with $V = (X, F)$, factors $\mathcal{F} = \{f_\alpha\}_{\alpha \in F}$, elimination order $o = [v_1, \cdots, v_n]$ and bound on bucket size $ibound$.
$\mathcal{F}' \leftarrow \mathcal{F}$
$\bar{o} \leftarrow \emptyset, \bar{w} \leftarrow \emptyset$.
Initialize by $\bar{o} = \emptyset, \bar{\mathbf{w}} = \emptyset$.
for $v$ in $o$ do
    $B_v \leftarrow \{f_\alpha | f_\alpha \in \mathcal{F}', v \in N(\alpha)\}$
    Partition $B_v$ into $R_v$ subgroups $\{B_v^r\}_{r=1}^{R_v}$ such that $|var(B_v^r)| \leq ibound$ for all $r$.
    Assign weights $w_1, \cdots, w_{R_v}$ while satisfying $\sum_r w_r = 1$.
    for $r \leftarrow 1, \cdots, R_v$ do
        Generate a new factor by:
            $f_{B_v^r}(\mathbf{x}_{B_v^r}) = \sum^{w_r}_{x_v} \prod_{f_\alpha \in B_v^r} f_\alpha(\mathbf{x}_\alpha), \ \forall\, \mathbf{x}_{B_v}.$
        $\mathcal{F}' \leftarrow \mathcal{F}' \cup \{f_{B_v^r}\} - B_v^r$
    end for
    Extend $\bar{o}$ by $[v^{(1)}, \cdots, v^{(R_v)}]$
    Extend $\bar{\mathbf{w}}$ by $[w_1, \cdots, w_{R_v}]$
end for
Initialize by $G_{v\alpha} = \mathbb{I}$ for all $(v, \alpha) \in E$.
for $t = 1, 2, \cdots, T$ do
    for $v$ in $X$ with $N(v) = \{\alpha, \beta\}$ do
        Update $G_{v\alpha}$ by (10).
        $G_{v\beta} \leftarrow (G_{v\alpha}^\top)^{-1}$
        Set $f_\alpha(\mathbf{x}_\alpha) \leftarrow \widehat{f}_\alpha(\mathbf{x}_\alpha; \mathbf{G}_\alpha)$ and $f_\alpha(\mathbf{x}_\alpha) \leftarrow \widehat{f}_\alpha(\mathbf{x}_\alpha; \mathbf{G}_\alpha)$ for all $\mathbf{x}_\alpha, \mathbf{x}_\beta$.
        Reset gauges $G_{v\alpha}, G_{v\beta} \leftarrow \mathbb{I}$.
    end for
end for
Output: $Z_{\text{WMBE-G}} = \prod_{f' \in \mathcal{F}'} f$
```

3.3 Relation to Previous Work

Hölder's inequality holds even for negative-valued functions, so we do not need to put any additional constraint on non-negativity of factors, e.g., $\widehat{f}_\alpha(\bar{\mathbf{x}}_\alpha; \mathbf{G}_\alpha) \geq 0$. Thus, invalid gauged transformed GMs are allowed for (9). This contrasts with the earlier work of [26], where additional non-negativity constraints were needed to restrict the gauge transformations considered. Consequently, to our knowl-

[2] See Section 4 for details of our choice of step size.

edge, our formulation is the first to explore the full range of freedom in gauge transformations when combined with methods of variational inference for GMs. Further, avoiding these non-negativity constraints simplifies our optimization procedure enabling an approach which scales much better than that of [26].

We emphasize that our optimization formulation (9) is a strict generalization of the approach of [16] which optimizes the WMBE bound with respect to *reparameterization* of GMs. Specifically, the GM reparameterized with respect to reparameterization parameters $\boldsymbol{\theta} = [\theta_{v\alpha} : (v,\alpha) \in E]$ consists of factors:

$$\widehat{f}_\alpha(\mathbf{x}_\alpha;\boldsymbol{\theta}_\alpha) = \prod_{v\in N(\alpha)} \exp(\theta_{v\alpha}(x_i)) f_\alpha(\mathbf{x}_\alpha), \qquad (12)$$

where $\boldsymbol{\theta}_\alpha = \{\theta_{v\alpha} : v \in N(\alpha)\}$. Here, the reparameterization parameter $\theta_{v\alpha}$ is constrained to satisfy the following constraint:

$$\exp(\theta_{v\alpha}(x_v) + \theta_{v\beta}(x_v)) = 1 \quad \forall\, v \in X, x_v, \qquad (13)$$

where $N(v) = \{\alpha, \beta\}$. With this constraint, it is easy to check that such transformations are distribution-invariant [18] and form a strict subset of gauge transformations. Alternatively, when gauges are restricted to diagonal matrices with non-negative elements, (2) and (3) match (12) and (13), respectively. Therefore, optimizing (9) is guaranteed to perform no worse than that of [16]. Formally, we provide the following analytic class of GMs where gauge transformations are expected to perform strictly better than reparameterizations. Here, we say a function of binary variables is *symmetric* if its value is invariant under a 'flipping' of all variables in its scope, e.g., $f_\alpha(2,1,2) = f_\alpha(1,2,1)$.

Theorem 1. *Consider a GM over binary variables (i.e., $d = 2$) where every factor f_α is symmetric. Then, $\boldsymbol{\theta} = \{\theta_{v\alpha}(x_v) = 0,\ \forall (v,\alpha) \in E, x_v\}$ is always a solution of the following optimization:*

$$\begin{aligned} &\textit{minimize}_{\boldsymbol{\theta}} && \sum_{\bar{x}_{\bar{1}}}^{\bar{w}_{\bar{n}}} \cdots \sum_{\bar{x}_1}^{\bar{w}_1} \prod_{\alpha\in F} \widehat{f}_\alpha(\bar{\mathbf{x}}_\alpha;\boldsymbol{\theta}_\alpha), \\ &\textit{subject to} && \exp(\theta_{v\alpha}(x_v) + \theta_{v\beta}(x_v)) = 1 \quad \forall\, v \in X, x_v. \end{aligned}$$

The proof of Theorem 1 is given in the Supplement. It shows that for symmetric GMs, e.g., the Ising model with no magnetic field, reparameterization is impossible to improve the WMBE bound. On the other hand, gauges are expected to improve it as we explain in what follows. We first remark that the optimality condition for reparameterization is equivalent to the zero gradient condition for diagonal elements of gauges, i.e., $\sum_{\bar{\mathbf{x}}_{\alpha\setminus v}} q(\bar{\mathbf{x}}_\alpha) = \sum_{\bar{\mathbf{x}}_{\beta\setminus v}} q(\bar{\mathbf{x}}_\beta)$, which aims to match the auxiliary marginals of variables split by WMBE. Under symmetric models, variables are indistinguishable from an auxiliary marginals point of view, which leads to Theorem 1. On the other hand, the zero gradient condition for non-diagonal gauges is harder to match since it takes local conditional dependency into account, e.g., considers $f_\alpha(\bar{\mathbf{x}}_{\alpha\setminus v}, x'_v)/f_\alpha(\bar{\mathbf{x}}_{\alpha\setminus v}, x''_v)$ upon evaluating the gradient. For symmetric GMs, the above reasoning for reparameterization fails since variables are distinguishable after conditioning, e.g., $f_\alpha(\bar{\mathbf{x}}_{\alpha\setminus v}, x'_v) \neq f_\alpha(\bar{\mathbf{x}}_{\alpha\setminus v}, x''_v)$. Namely, optimal gauges believably have non-diagonal elements. Indeed, in all our experiments, gauge transformations significantly outperform reparameterizations.

4 EXPERIMENTS

In this section, we report experimental results on performance of our proposed algorithms for the task of upper bounding the partition function Z.

4.1 Setup

Experiments were conducted with three family of GMs: (i) Ising models on a 10 × 10 grid graph (non-toroidal) with 180 factors/100 variables; (ii) Forney-style GMs on the 3-regular graph with 180 factors/270 variables; and (iii) Linkage dataset from UAI 2014 Inference Competition [34].

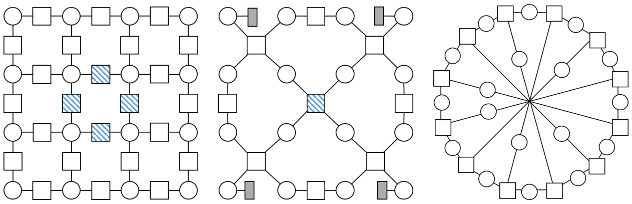


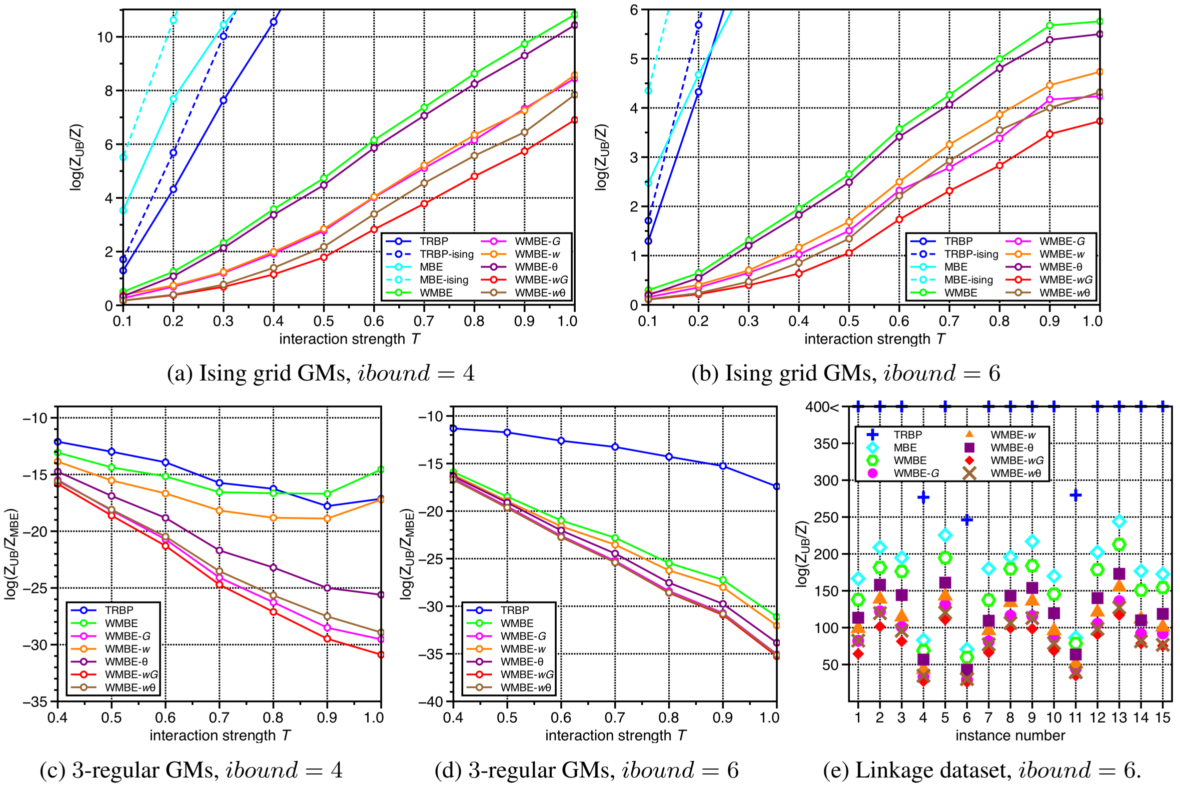
Figure 3: Illustration of Ising grid GM (left), its equivalent Forney-style GM (middle) and 3-regular graph (right) of interest. Factors surrounding the selected lattice (blue, dashed) are contracted into a single factor, and then uniform single potentials (grey, filled) are added for variables of degree 1.

(i) Ising models. Ising models were defined with mixed interactions (spin glasses):

$$p(\mathbf{x}) = \frac{1}{Z} \exp\Big(\sum_{v\in X} \phi_v x_v + \sum_{(u,v)\in E} \phi_{uv} x_u x_v\Big),$$

where $x_u \in \{-1,1\}$ and $\phi_v \sim \mathcal{N}(0, 0.1)$, $\phi_{uv} \sim \mathcal{N}(0, T)$. Here, $T \geq 0$ is the 'interaction strength' parameter that controls the degree of interactions between variables. When $T = 0$, variables are independent. As T grows, the inference task is typically harder.

Note that this Ising model is not in Forney-style form, with variables adjacent to at most 4 pairwise factors. Hence, to apply our gauge optimization framework, we generate an equivalent Forney-style GM using the transformation introduced in [30]: this maps any classical lattice model (allow-



(a) Ising grid GMs, $ibound = 4$

(b) Ising grid GMs, $ibound = 6$

(c) 3-regular GMs, $ibound = 4$

(d) 3-regular GMs, $ibound = 6$

(e) Linkage dataset, $ibound = 6$.

Figure 2: Performance comparisons in various families of GMs.

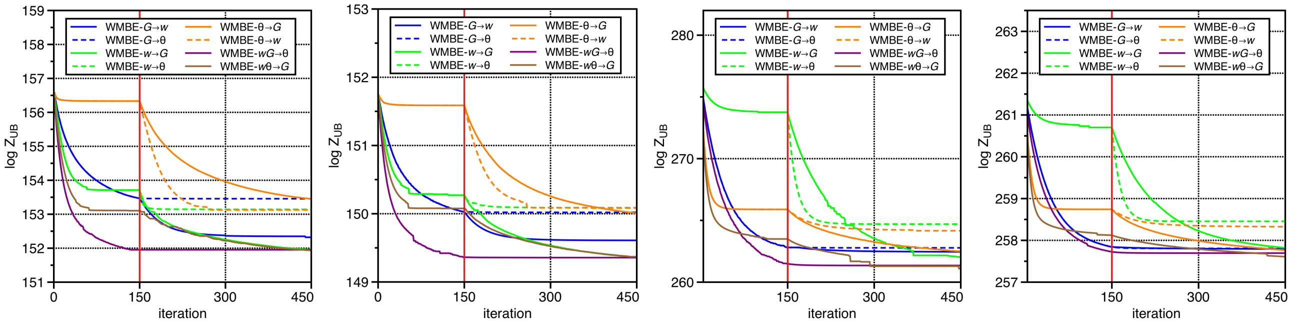
ing for magnetic fields/singleton potentials) into an equivalent Forney-style model. At a high level, the transformation chooses disjoint lattices to cover the whole graph, then contracts each lattice into a single factor. Levin and Nave [30] showed that one can always choose the lattice smartly so that each vertex is covered exactly twice, resulting in a Forney-style GM (see Figure 3 for details). Notably, this GM has relatively low induced width of 14, thus the partition function can be computed exactly in reasonable time (though still computationally hard) by using BE.

(ii) 3-regular Forney-stlye GMs. We considered 3-regular Forney-style GMs with log-factors drawn from normal distribution, i.e., $\log f_\alpha(\mathbf{x}_\alpha) \sim \mathcal{N}(0, T)$. Again, $T \geq 0$ is the interaction strength parameter. In this case, we would like to choose graphs so that the induced width is high and the partition function is hard to compute. To this end, we aligned factors in a cycle, and assigned variables (edges) between adjacent factors in the cycle as well as those in the opposite side if it. See Figure 3 for its illustration. This choice gives high induced width, e.g., naïvely applying BE by eliminating variables between adjacent factors in clockwise elimination order results in bucket size $2^{|V|/2+2}$.

(iii) UAI Linkage dataset. Finally, we consider a family of real-world models from the UAI 2014 Inference Competition, namely the Linkage (genetic linkage) dataset. Specifically, the family consists of GMs with average of 949.94 variables with averaged maximum cardinality $\max_{i\in\mathcal{V}} |\mathcal{X}_i| = 4.95$ and 727.35 non-singleton hyperedges with averaged maximum size $\max_{\alpha\in\mathcal{E}} |\alpha| = 4.47$. Since GMs in Linkage dataset were not of Forney-style form, we constructed an equivalent Forney-style GM as in Figure 1.

Comparing approaches. We compared our gauged algorithm WMBE-G, i.e. optimizing the WMBE bound jointly with gauges and Hölder weights, to earlier methods considered in [16]: the unoptimized WMBE bound (‘WMBE’), its optimized versions with respect to Hölder weights w and/or reparameterizations θ (‘WMBE-w’, ‘WMBE-θ’ and ‘WMBE-$w\theta$’). Further, we also ran the following popular baselines for computing upper bounds on Z: standard mini-bucket elimination (‘MBE’) and tree re-weighted belief propagation (‘TRBP’) [12]. Finally, for fair comparisons in Ising grid GMs, we additionally compared to MBE and TRBP run on the original Ising grid GM (MBE-Ising and TRBP-Ising) in order to validate whether the forementioned GM transformation to a Forney-style model is ‘favored’ towards gauge optimization.

Further details. Hölder weights w and reparameterizations θ were updated using projected gradients and log-gradients respectively, as proposed in [16]. Step sizes for gradients were chosen as $0.01, 0.1, 0.1$ for optimizing each of gauge, Hölder weights, and reparameterizations, respectively. These were chosen empirically for ‘easy’ convergence in our experiments – there exists room for tuning



(a) Ising grid GMs, $ibound = 4$ (b) Ising grid GMs, $ibound = 6$ (c) 3-regular GMs, $ibound = 4$ (d) 3-regular GMs, $ibound = 6$

Figure 4: Effectiveness of optimizing various parameter choices (all methods return an upper bound on $\log Z$).

or for more sophisticated gradient descent methods such as [35]. TRBP was run with damping until convergence. For Ising grid GMs, we measure the log-error (with base e) approximating the partition function Z, i.e., $\log(Z_{\text{UB}}/Z)$ where Z_{UB} is the upper bound of a respective algorithm. For 3-regular GMs, it is impossible to measure (since Z is impossible to compute), and instead we use the relative magnitude of bounds with respect to the mini-bucket upper bound Z_{MBE}, i.e., $\log(Z_{\text{UB}}/Z_{\text{MBE}})$. Since all tested algorithms provide guaranteed upper bounds on Z, a lower number indicates better performance. Further, in the UAI dataset, 2 out of 17 instances were omitted since it had factors with size larger than the algorithm's $ibound$ of our choice. Finally, each point in the plots represents results averaged over 10 independent runs.

4.2 Experimental Results

As shown in Figures 2(a)-(b), TRBP and MBE perform better on the transformed Forney-style GMs than on the original Ising models (this may be interesting to explore in future work), but not by nearly enough to achieve the performance of the other methods. For fair comparison, we should examine the 'TRBP' and 'MBE' plots rather than the '-Ising' versions. We observe that WMBE-wG, which enjoys the most freedom in optimization of the WMBE bound, outperforms all other tested algorithms. In particular, the benefit of WMBE-wG appears to increase with higher interaction strength. Comparing optimizations of just one class of parameters, i.e., WMBE-G, WMBE-w, WMBE-θ, we observe that WMBE-G performs at least as well as others. In particular, optimizing gauges is always better than optimizing over the subclass of reparameterizations, i.e., WMBE-G and WMBE-wG always outperform WMBE-θ and WMBE-$w\theta$, respectively. Further, WMBE-G outperforms other approaches significantly for 3-regular GMs and UAI dataset, where it outperforms even WMBE-$w\theta$ in 3-regular GMs with $ibound = 4$ and some instances of the UAI dataset.

Next, we consider experiments on specific instances of the Ising grid GM and 3-regular GM with $T = 1.0$ in order to measure the effectiveness of optimizing each parameter G, w, θ separately over iterations; see Figure 4. Specifically, we first optimize a chosen parameter with respect to WMBE related bounds (via gradient descent methods) for an initial 150 iterations. Then, we change the parameter to optimize further (e.g., $G \to \theta$) for another 300 iterations to observe the additional benefit from optimizing the second parameter. The running times per iteration for all parameters are comparable. We observe that G methods perform very well, which is particularly impressive since we use a small step size for gauges. Overall, observed performance gains may be ranked as: gauges $>$ weights $>$ reparameterization for Ising grid GMs; and gauges $>$ reparameterization $>$ weights for 3-regular GMs. Gauge optimization is critical for the best performance in all experiments. As expected, wG yields the best results. For 3-regular GMs, gauge optimization alone is almost optimal.

5 Conclusion

We developed a new gauge-variational approach to yield guaranteed bounds on the partition functions of GMs by jointly optimizing variational parameters and gauge transformations. Our approach has better scaling characteristics then other recent state-of-the-art methods, and should be of significant practical value.

Acknowledgements

AW acknowledges support from the David MacKay Newton research fellowship at Darwin College, The Alan Turing Institute under EPSRC grant EP/N510129/1 & TU/B/000074, and the Leverhulme Trust via the CFI. This work was partly supported by the ICT R&D program of MSIP/IITP [R-20161130-004520, Research on Adaptive Machine Learning Technology Development for Intelligent Autonomous Digital Companion]. The work of MC at LANL was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy under Contract No. DE-AC52-06NA25396.

References

[1] Robert Gallager. Low-density parity-check codes. *IRE Transactions on information theory*, 8(1):21–28, 1962.

[2] Frank R. Kschischang and Brendan J. Frey. Iterative decoding of compound codes by probability propagation in graphical models. *IEEE Journal on Selected Areas in Communications*, 16(2):219–230, 1998.

[3] Hans A. Bethe. Statistical theory of superlattices. *Proceedings of Royal Society of London A*, 150:552, 1935.

[4] Rudolf E. Peierls. Ising's model of ferromagnetism. *Proceedings of Cambridge Philosophical Society*, 32:477–481, 1936.

[5] Marc Mézard, Georgio Parisi, and M. A. Virasoro. *Spin Glass Theory and Beyond*. Singapore: World Scientific, 1987.

[6] Giorgio Parisi. Statistical field theory, 1988.

[7] Marc Mezard and Andrea Montanari. *Information, Physics, and Computation*. Oxford University Press, Inc., New York, NY, USA, 2009.

[8] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

[9] Michael I. Jordan. *Learning in graphical models*, volume 89. Springer Science & Business Media, 1998.

[10] William T. Freeman, Egon C. Pasztor, and Owen T. Carmichael. Learning low-level vision. *International journal of computer vision*, 40(1):25–47, 2000.

[11] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM Journal on computing*, 22(5):1087–1116, 1993.

[12] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51(7):2313–2335, 2005.

[13] Judea Pearl. *Reverend Bayes on inference engines: A distributed hierarchical approach*. Cognitive Systems Laboratory, School of Engineering and Applied Science, University of California, Los Angeles, 1982.

[14] Qiang Liu and Alexander Ihler. Negative tree reweighted belief propagation. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pages 332–339. AUAI Press, 2010.

[15] Stefano Ermon, Ashish Sabharwal, Bart Selman, and Carla P. Gomes. Density propagation and improved bounds on the partition function. In *Advances in Neural Information Processing Systems*, pages 2762–2770, 2012.

[16] Qiang Liu and Alexander T. Ihler. Bounding the partition function using Hölder's inequality. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 849–856, 2011.

[17] Alexander Novikov, Anton Rodomanov, Anton Osokin, and Dmitry Vetrov. Putting MRFs on a tensor train. In *International Conference on Machine Learning*, pages 811–819, 2014.

[18] Martin J. Wainwright, Tommy S. Jaakkola, and Alan S. Willsky. Tree-based reparametrization framework for approximate estimation on graphs with cycles. *Information Theory, IEEE Transactions on*, 49(5):1120–1146, 2003.

[19] Michael Chertkov and Vladimir Chernyak. Loop calculus in statistical physics and information science. *Physical Review E*, 73:065102(R), 2006.

[20] Michael Chertkov and Vladimir Chernyak. Loop series for discrete statistical models on graphs. *Journal of Statistical Mechanics*, page P06009, 2006.

[21] Leslie G. Valiant. Holographic algorithms. *SIAM Journal on Computing*, 37(5):1565–1594, 2008.

[22] Ali Al-Bashabsheh and Yongyi Mao. Normal factor graphs and holographic transformations. *IEEE Transactions on Information Theory*, 57(2):752–763, 2011.

[23] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1):1–305, 2008.

[24] G. David Forney Jr and Pascal O. Vontobel. Partition functions of normal factor graphs. *arXiv preprint arXiv:1102.0316*, 2011.

[25] Michael Chertkov. Lecture notes on "statistical inference in structured graphical models: Gauge transformations, belief propagation & beyond", https://sites.google.com/site/mchertkov/courses, 2016.

[26] Sung-Soo Ahn, Michael Chertkov, and Jinwoo Shin. Gauging variational inference. In *Advances in Neural Information Processing Systems 30*, pages 2881–2890. Curran Associates, Inc., 2017.

[27] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

[28] Rina Dechter and Irina Rish. Mini-buckets: A general scheme for bounded inference. *Journal of the ACM (JACM)*, 50(2):107–153, 2003.

[29] G. David Forney. Codes on graphs: Normal realizations. *IEEE Transactions on Information Theory*, 47(2):520–548, 2001.

[30] Michael Levin and Cody P. Nave. Tensor renormalization group approach to two-dimensional classical lattice models. *Physical review letters*, 99(12):120601, 2007.

[31] Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113(1):41–85, 1999.

[32] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[33] G.H. Hardy, J.E. Littlewood, and G. Pólya. Inequalities, 1934.

[34] Vibhav Gogate. UAI 2014 Inference Competition. http://www.hlt.utdallas.edu/~vgogate/uai14-competition/index.html, 2014.

[35] Yurii Nesterov. A method of solving a convex programming problem with convergence rate o(1/k2). *Soviet Mathematics Doklady*, 27(2):372–376, 1983.