# Probabilistic Inference with Algebraic Constraints: Theoretical Limits and Practical Approximations

**Zhe Zeng**[*]
CS Department
UCLA
zhezeng@cs.ucla.edu

**Paolo Morettin**[*†]
DTAI
KU Leuven, Belgium
paolo.morettin@unitn.it

**Fanqi Yan**[*]
CS Department
UT Austin
fqyan@cs.utexas.edu

**Antonio Vergari**
CS Department
UCLA
aver@cs.ucla.edu

**Guy Van den Broeck**
CS Department
UCLA
guyvdb@cs.ucla.edu

## Abstract

Weighted model integration (WMI) is a framework to perform advanced probabilistic inference in hybrid domains, i.e., on distributions over mixed continuous-discrete random variables and in the presence of complex logical and arithmetic constraints. In this work, we advance the WMI framework on both the theoretical and algorithmic side. First, we trace the boundaries of tractability for WMI inference in terms of two key properties of a WMI problem's dependency structure: sparsity and diameter. We prove that exact inference is only efficient if that structure is tree-shaped with logarithmic diameter. While this result deepens our theoretical understanding of WMI it hinders the practical applicability of exact WMI solvers to large problems. To overcome this, we propose the first approximate WMI solver that does not resort to sampling, but performs exact inference on an approximate model. Our solution iteratively performs message passing in a relaxed problem structure to recover lost dependencies. As our experiments show, it scales to problems that are out of the reach of exact WMI solvers while delivering accurate approximations.

## 1 Introduction

Consider an autonomous agent operating under uncertainty in a real-world scenario, for instance a self-driving vehicle. It has to model both *continuous* variables like the speed and position of other cars and *discrete* ones like the color of traffic lights and the number of pedestrians. Moreover, in order to make decisions, it needs to perform advanced probabilistic reasoning. For example, it has to reason about physical constraints while computing the probability of a grounded scene described via complex *algebraic constraints*, such as the geometry of vehicles and the roads ahead.

Performing probabilistic inference in these constrained and hybrid (mixed continuous-discrete) scenarios goes beyond the limited inference capabilities of intractable probabilistic models such as variational autoencoders [28] and generative adversarial networks [25]. This is also the case for classical probabilistic graphical models for hybrid domains [27, 32] and more recent tractable alternatives [33, 38, 40] which struggle to either perform inference over complex algebraic constraints or make too simplistic representational or distributional assumptions.

[*]Authors contributed equally. This research was performed while F.Y. and P.M. were visiting UCLA.
[†]This work was partially carried out when P.M. was working at the University of Trento.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

On the other hand, Weighted Model Integration (WMI) [8, 34] is a modeling and inference framework that supports general hybrid probabilistic reasoning over algebraic constraints, *by design*. Indeed, in the WMI framework, mixed complex continuous-discrete interactions can be easily expressed in the language of Satisfiability Modulo Theories (SMT) [7] and answering probabilistic queries involving algebraic constraints can be naturally cast as integration of certain weight functions over the regions that satisfy those constraints.

In this paper we advance the WMI framework on two fronts. First, we deepen the theoretical understanding of the complexity of WMI inference on real-world problems by proving hardness results. Second, we deliver an efficient and accurate approximate WMI solver as a practical algorithmic solution to deploy WMI inference at a larger scale.

Specifically, we study the dependency structure of WMI problems as specified by the primal graph of their SMT formula [22]. We prove that performing exact inference is #P-hard if the primal graph has a treewidth larger than one or a diameter that is linear in the number of variables. Second, to overcome these negative results, we introduce ReCoIn , a practical algorithmic solution that extends the relax-compensate-and-recover framework [14, 16, 17] for approximate discrete inference to hybrid inference scenarios with algebraic constraints. As our experiments suggest ReCoIn candidates as the best alternative, in terms of scalability and accuracy of the delivered approximations, in the current panorama of general-purpose WMI solvers.

The rest of the paper is organized as follows. In Section 2 we introduce the notation and background needed to later prove our theoretical results in Section 3 and to introduce ReCoIn in Section 4. Before evaluating ReCoIn in Section 6 we discuss related work in Section 5.

## 2 Background

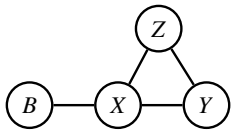
**Notation.** Uppercase letters denote random variables ($X, B$) and lowercase letters denote their assignments ($x, b$). We use bold for sets of variables ($\mathbf{X}, \mathbf{B}$), and their joint assignments ($\boldsymbol{x}, \boldsymbol{b}$). We use capital Greek letters for logical formulas ($\Gamma, \Delta$). Literals are atomic formulas or their negation, and are denoted using either $\ell$ or lowercase Greek letters ($\gamma, \delta$). We let $\boldsymbol{x} \models \Delta$ denote the satisfaction of a formula $\Delta$ by an assignment $\boldsymbol{x}$. Its corresponding indicator function is $[\![\boldsymbol{x} \models \Delta]\!]$.

**Satisfiability Modulo Theories .** To represent complex relationships between discrete and continuous variables, we harness the language of Satisfiability Modulo Theories (SMT) [7] which generalizes Boolean propositional logic [6]. Specifically, we use SMT over linear real arithmetic ($\mathcal{LRA}$) which has been used as an expressive modeling language for probabilistic programming [13], model checking [23] and robotics [20]. As is common, we adopt quantifier-free SMT($\mathcal{LRA}$) formulas and we assume them to be in conjunctive normal form (CNF), that is, a conjunction of clauses. For brevity, we will refer to them as simply SMT formulas. To characterize the dependency structure of an SMT formula we make us of its *primal graph* representation.

**Definition 2.1.** *(Primal Graph) Let $\Delta$ be an SMT formula. Then its primal graph $\mathcal{G}_\Delta = (\mathcal{V}, \mathcal{E})$ is the undirected graph whose vertex set $\mathcal{V}$ is the set of variables in formula $\Delta$, and whose edge set $\mathcal{E}$ has edge $X - Y$ iff variable $X$ and variable $Y$ appear together in one clause $\Gamma \in \Delta$.*

**Example 2.2** (SMT formula and its primal graph)**.** *Consider the SMT formula $\Delta$ on the left over continuous variables $X, Y, Z$ and boolean variable $B$, its primal graph $\mathcal{G}_\Delta$ is shown on the right.*

$$\Delta = \begin{cases} (0 \leq X \leq 2) \wedge (1 \leq Y \leq 2) \wedge (0 \leq Z \leq 2) \\ (X \geq 1) \vee B \\ (X + Y \leq 3) \wedge (X + Z \geq 2) \wedge (Y + Z \leq 3) \end{cases}$$




**Weighted Model Integration (WMI).** Weighted Model Integration (WMI) [8, 34] is a framework for probabilistic modeling and inference over mixed continuous-discrete distributions in presence of algebraic constraints defined as SMT formulas. These representations are captured by WMI models.

**Definition 2.3.** *(WMI model) Let $\mathbf{X}$ be a set of continuous random variables assuming values in $\mathbb{R}$, and $\mathbf{B}$ a set of Boolean random variables assuming values in $\mathbb{B} = \{\texttt{true}, \texttt{false}\}$. A WMI model is a pair $(\Delta, w)$, where $\Delta$ is an SMT formula over $\mathbf{X}$ and $\mathbf{B}$, and $w : (\boldsymbol{x}, \boldsymbol{b}) \mapsto \mathbb{R}^+$ is a positive function, called the weight function.*

We consider classes of WMI problems whose weight function comes from a parametric function family, denoted $\mathbf{\Omega}$. Moreover, we adopt the common assumption of weight functions $w$ to be defined as products of per-literal weights [8, 11, 41]. That is, $w$ is definable via a set of functions $\mathcal{W} = \{w_\ell(\boldsymbol{x})\}_{\ell \in \mathcal{L}}$, where $\mathcal{L}$ are the literals in $\Delta$. and where each $w_\ell$ is defined over variables in literal $\ell$. Then, the weight of assignment $(\boldsymbol{x}, \boldsymbol{b})$ is: $w(\boldsymbol{x}, \boldsymbol{b}) = \prod_{\ell \in \mathcal{L}} w_\ell(\boldsymbol{x}, \boldsymbol{b})^{[\![\boldsymbol{x}, \boldsymbol{b} \models \ell]\!]}$. Hence, we will represent WMI models as pairs $(\Delta, \mathcal{W})$.

**Definition 2.4.** *(WMI task) Let $(\Delta, \mathcal{W})$ be a WMI model over real variables $\mathbf{X}$ and Boolean variables $\mathbf{B}$. The* WMI task *for $(\Delta, \mathcal{W})$ is to compute*

$$\mathsf{WMI}(\Delta, \mathcal{W}; \mathbf{X}, \mathbf{B}) \triangleq \sum_{\boldsymbol{b} \in \mathbb{B}^{|\mathbf{B}|}} \int_{(\boldsymbol{x}, \boldsymbol{b}) \models \Delta} \prod\nolimits_{\ell \in \mathcal{L}} w_\ell(\boldsymbol{x}, \boldsymbol{b})^{[\![\boldsymbol{x}, \boldsymbol{b} \models \ell]\!]} \, d\boldsymbol{x}. \tag{1}$$

*That is, the task is to sum over all possible Boolean assignments $\boldsymbol{b} \in \mathbb{B}^{|\mathbf{B}|}$ while integrating over the weighted assignments of $\mathbf{X}$ that satisfy the formula: $(\boldsymbol{x}, \boldsymbol{b}) \models \Delta$.*

When all weights $w_\ell(\boldsymbol{x})$ are constants and all variables continuous ($\mathbf{B} = \emptyset$) we retrieve the model integration (MI) task [41], whereas when all variables are Boolean (i.e., $\mathbf{X} = \emptyset$) WMI equals the well-known weighted model counting (WMC) task [11]. In the general case, solving $\mathsf{WMI}(\Delta, \mathcal{W}; \mathbf{X}, \mathbf{B})$ equals to computing the partition function of the unnormalized probability distribution induced by weights $\mathcal{W}$ on formula $\Delta$ and restricted to the regions where $\Delta$ is SAT.

As such, computing the probability of an event represented as an SMT formula $\Phi$ involving algebraic constraints w.r.t. the distribution induced by $\mathcal{W}$ on $\Delta$ can be done by computing the WMI of the conjunction of formula $\Delta$ and formula $\Phi$, normalized by the partition function:

$$\mathsf{Pr}_\Delta(\Phi) = \mathsf{WMI}(\Delta \wedge \Phi, \mathcal{W}; \mathbf{X}, \mathbf{B}) \;/\; \mathsf{WMI}(\Delta, \mathcal{W}; \mathbf{X}, \mathbf{B}).$$

**Example 2.5** (Advanced probabilistic inference with WMI)**.** *Consider the SMT formula $\Delta$ in Example 2.2 with per-literal weights $\mathcal{W} = \{w_{\ell_1}(B) := 2;\ w_{\ell_2}(x) := x^2;\ w_{\ell_3}(y, z) := 2yz\}$ where $\ell_1 := B$, $\ell_2 := x \geq 1$, $\ell_3 := y + z \leq 3$ and all the weights associated to other literals are constantly 1. Then the WMI of formula $\Delta$ evaluates to:*

$$\mathsf{WMI}(\Delta, \mathcal{W}; \mathbf{X}, B) = \int_1^2 dx \int_1^{-x+3} dy \int_{-x+2}^{-y+3} x^2 \cdot (2+1) \cdot (x+y) \cdot 2yz \, dz = \frac{11173}{480}.$$

*Moreover, for the two formulas $\Phi_c = (B = true)$ and $\Phi_1 = 0 \leq z \leq 1$, then*

$$\mathsf{Pr}_\Delta(\Phi_1 | \Phi_c) = \mathsf{WMI}(\Delta \wedge \Phi_c \wedge \Phi_1, \mathcal{W}; \mathbf{X}, B) \;/\; \mathsf{WMI}(\Delta \wedge \Phi_c, \mathcal{W}; \mathbf{X}, B) = 18936 \;/\; 78211 \approx 0.242.$$

From here on, w.l.o.g. we will assume WMI problems to be defined on continuous variables only. We leverage the polytime reduction introduced in Zeng and Van den Broeck [41] to map a WMI problem $(\Delta, \mathcal{W})$ over continuous and Boolean variables $\mathbf{X}$ and $\mathbf{B}$ to a new WMI problem $(\Delta', \mathcal{W}')$ over continuous variables $\mathbf{X}'$ only. This is done by properly introducing auxiliary variables in $\mathbf{X}'$ to account for $\mathbf{B}$. The resulting primal graph $\mathcal{G}_{\Delta'}$ is isomorphic to $\mathcal{G}_\Delta$. For instance, we can replace $B$ in Example 2.5 by a real variable $T_B$ having values in $[-1, 1]$ without changing the WMI task nor the treewidth or the diameter of the primal graph (cf. Appendix A).

## 3 On the hardness of WMI

While the general formulation of WMI we have provided in the previous section is elegant and appealing for advanced probabilistic reasoning, it is, however, not practical in general. In fact, it requires solving an arbitrarily complex integral, which is a #P-hard problem [5].

To fill this gap, recent works have started looking for *classes of tractable WMI problems*, i.e., problems for which a solution can be computed exactly in polytime [41, 42]. These classes of problems can be characterized by two parameters: the *treewidth* and the *diameter* of the primal graph of the SMT formulas considered, where the latter is generally expressed as a function of the number of variables in the problem. Note that this is strikingly different from classical discrete probabilistic graphical models, where most of the complexity results are stated in terms of the treewidth alone [31, 36].

**Definition 3.1.** *($\mathcal{WMI}(\mathbf{\Omega}, \delta, t)$ Problem Class) Let $\mathcal{WMI}(\mathbf{\Omega}, \delta, t)$ be the class of WMI problems over models of the form $(\Delta, \mathcal{W})$ on real domains, having primal graph $\mathcal{G}_\Delta$ with diameter of $\Theta(\delta(n))$ and treewidth $t$, where $n$ is the number of variables in the formula $\Delta$; and having per-literal weights $\mathcal{W}$ in a function family $\mathbf{\Omega}$.*

The largest tractable WMI class known so far has been introduced in Zeng et al. [42] as $\mathcal{WMI}(\mathbf{\Omega}, \log(n), 1)$, i.e., the class of problems over $n$ real variables whose primal graph is tree-shaped (treewidth 1) and has diameter of length logarithmic in $n$, and whose weight functions belong to a function family $\mathbf{\Omega}$ satisfying some conditions called *tractable weight conditions* (TWCs).

**Definition 3.2.** *(TWCs) Given a parametric weight function family $\mathbf{\Omega}$, it satisfies the TWCs iff*

*i) it is closed under product, i.e., $\forall f, g \in \mathbf{\Omega},\ f \cdot g \in \mathbf{\Omega}$;*
*ii) it is closed under definite integration, i.e., $\forall f \in \mathbf{\Omega}, F(u(x)) - F(l(x)) \in \mathbf{\Omega}$ where $F$ is the antiderivative of $f$, and $l(x), u(x)$ are SMT($\mathcal{LRA}$) integration bounds for any $x \in \mathbf{X}$;*
*iii) the symbolic antiderivative of any $f \in \mathbf{\Omega}$ can be tractably computed by symbolic integration.*

Examples of weight functions in family $\mathbf{\Omega}$ include the largely adopted family of (piecewise) polynomials [8], the family of exponentiated linear functions and the family of their products. In the following analysis, we will restrict our attention to weight function families satisfying the TWCs.

In Zeng et al. [42] the tractability of problem class $\mathcal{WMI}(\mathbf{\Omega}, \log(n), 1)$ is demonstrated by construction, where they introduce a message passing scheme, named MP-WMI, that runs in polytime on tree-shaped and diameter-bounded primal graphs. That is, some *sufficient* conditions for tractable WMI classes are provided. Here we provide a finer charting of the "tractable islands" of WMI problems by questioning the necessity of the above conditions while looking for larger tractable classes. We prove that unless $P = NP$, larger classes are not tractable. We begin by proving that increasing the diameter of a tree-shaped problem structure makes it hard.

**Theorem 3.3.** *Let $\mathcal{WMI}(\mathbf{\Omega}, n, 1)$ be the class of WMI problems whose weight function family $\Omega$ satisfies the TWCs. Then inference in $\mathcal{WMI}(\mathbf{\Omega}, n, 1)$ is #P-hard.*

*Sketch of proof.* We build a polytime reduction from a #P-complete variant of the subset sum problem [24, 12, 26] to a WMI problem with constant weights and whose primal graph $\mathcal{G}_\Delta$ is a chain with diameter exactly $n$. A complete proof is in Appendix B. □

Next, we turn our attention to another class of WMI problems, the class $\mathcal{WMI}(\mathbf{\Omega}, \log(n), 2)$, having logarithmic diameter but treewidth 2. This class is also supposed to be "easy" in the sense that it extends the tractable class $\mathcal{WMI}(\mathbf{\Omega}, \log(n), 1)$ by slightly increasing the treewidth by one. Unfortunately, inference in $\mathcal{WMI}(\mathbf{\Omega}, \log(n), 2)$ is also hard.

**Theorem 3.4.** *Let $\mathcal{WMI}(\mathbf{\Omega}, \log(n), 2)$ be the class of WMI problems whose parametric weight function family $\Omega$ satisfies the TWCs. Then inference in $\mathcal{WMI}(\mathbf{\Omega}, \log(n), 2)$ is #P-hard.*

*Sketch of proof.* Analogously to Theorem 3.3, we prove it by constructing a polytime reduction from a #P-complete variant of the subset sum problem to a $\mathsf{MI}$ problem whose primal graph has treewidth two but diameter being at most $\log(n)$. A complete proof is provided in Appendix B. □

Note that our result differs from the one presented in [42] for the hardness of the class $2\mathsf{WMI}(\mathbf{\Omega})$, containing WMI problems with SMT formulas being conjunctions of clauses comprising at most two variables. In fact, $\mathcal{WMI}(\mathbf{\Omega}, \log(n), 2)$ is contained in $2\mathsf{WMI}(\mathbf{\Omega})$. As such, we trace the tractablity boundaries of WMI inference with higher precision, as the next corollary states. Its proof follows from Theorems 3.3 and 3.4 and from the sufficiency as demonstrated in Zeng et al. [42].

**Corollary 3.5.** *Let $\mathcal{WMI}(\mathbf{\Omega}, \log(n), t)$ be the class of WMI problems whose parametric weight function family $\Omega$ satisfies the TWCs. Then $\mathcal{WMI}(\mathbf{\Omega}, \log(n), t)$ is a tractable WMI class for inference if-and-only-if treewidth $t = 1$.*

These complexity results set the standard for the solver complexity: every exact WMI solver that aims to be efficient, needs to operate in the regime of Corollary 3.5. However, real-world problems do not always conform to the structural desiderata for primal graphs stated in it. This implies that efficient approximations might not only be useful in these scenarios, but *needed*. In the next section we fill this gap, by introducing our approximate WMI solver that navigates the tractable islands in WMI problems by performing efficient inference on a relaxed version of intractable WMI problems.

## 4 ReCoIn: Relax, compensate and then integrate

Our algorithm to approximate WMI inference comprises three phases: i) *RElaxing* an intractable WMI model into a simpler one amenable to exact inference by removing dependencies from it; then ii) introduce certain literals and weights to *COmpensate* for the dependency structure lost in this way and iii) optimize them by solving a series of exact *INtegration* problems. We name it ReCoIn. With ReCoIn we can navigate *a spectrum of approximations* — with the original primal graph $\mathcal{G}_\Delta$ on one end, and a fully disconnected version on the other — by removing more and more edges. As such, ReCoIn can be viewed as extending the *relax-compensate-recover* (RCR) framework [14, 16, 17] for approximate inference on discrete probabilistic models to continuous representations and in presence of algebraic constraints.

### 4.1 Relaxation: introducing and then "breaking" equivalence constraints

The aim of the relaxation step is to obtain a new SMT formula $\Delta^{\mathsf{rel}}$ such that its associated primal graph $\mathcal{G}_{\Delta^{\mathsf{rel}}}$, serves as the simplification of the original $\mathcal{G}_\Delta$ by removing a given set of edges. We will show that the removal of any edge can be formulated as the removal of an equivalence edge [17]. This process consists of two steps. First, we create an *augmented formula* $\Delta^{\mathsf{aug}}$ by introducing new variables to $\Delta$ and enforcing them to act as *copies* of certain original variables by explicitly adding *equivalence constraints*. Second, we deliver the relaxed $\mathcal{G}_{\Delta^{\mathsf{rel}}}$ by removing these equivalence constraints.

**Augmentation.** The detailed process of distilling a new augmented model $(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}})$ from $(\Delta, \mathcal{W})$, given a subset of edges $\mathcal{E}_d \subseteq \mathcal{E}$ in $\mathcal{G}_\Delta$ to remove, is listed in Algorithm 2 in Appendix C. At its core, there are routines for copying one variable and adding the corresponding equivalence constraints and compensating literals. For each edge $X_i - X_j \in \mathcal{E}_d$ to be removed, one of its variables is arbitrarily selected, say $X_i$. Then a variable $X_i^c$, as a copy of the chosen $X_i$, is introduced in $\Delta^{\mathsf{aug}}$ as well as one equivalence constraint between the two as the literal $\hat{\ell} : (X_i^c = X_i)$ with associated weight function $\delta(X_i, X_i^c)$ where $\delta$ is the Dirac delta function. Then we properly rename all occurrences of $X_i$ by $X_i^c$ in the literals appearing in the clauses of $\Delta^{\mathsf{aug}}$ that also contain $X_j$ and introduce copied literals for the univariate clauses over $X_i$ only. These steps cause the primal graph $\mathcal{G}_{\Delta^{\mathsf{aug}}}$ to now contain the dependency $X_i - X_i^c - X_j$ but not $X_i - X_j$.

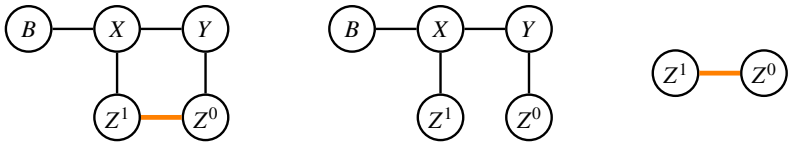
Note that the augmented WMI model $(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}})$ now contains more variables than the original one. Specifically, for each variable $X_i \in \mathcal{G}_\Delta$ we might have introduced $C_i$ different copies in $\mathcal{G}_{\Delta^{\mathsf{aug}}}$, denoted as $X_i^1, \ldots, X_i^{C_i}$, if we removed $C_i$ edges over $X_i$. We will denote the original $X_i$ as $X_i^0$ for notation consistency. Even if the dimensionality of the augmented WMI problem is increased by augmentation, the next propositions are guaranteeing that we are not altering the partition function and the marginal distributions of $\mathsf{Pr}_\Delta$, and that introducing equivalence constraints does not alter the induced distribution.

**Proposition 4.1.** *Let $\Delta$ be an SMT formula with primal graph $\mathcal{G}_\Delta$ and per-literal weight functions $\mathcal{W}$, and let $\Delta^{\mathsf{aug}}$ and $\mathcal{W}^{\mathsf{aug}}$ be the output of Algorithm 2 when applied to $\Delta$ and $\mathcal{G}_\Delta$ given a certain subset of edges in $\mathcal{G}_\Delta$. Then it holds that $\mathsf{WMI}(\Delta, \mathcal{W}) = \mathsf{WMI}(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}})$. Moreover, for any $X_i$ in $\mathcal{G}_\Delta$ and univariate literal $\ell$ over $X_i$, it holds that $\mathsf{Pr}_\Delta(\ell) = \mathsf{Pr}_{\Delta^{\mathsf{aug}}}(\ell)$.*

**Removing equivalence constraints.** Given an augmented model $(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}})$, we remove equivalence constraints introduced at the augmentation step to obtain the relaxed model $(\Delta^{\mathsf{rel}}, \mathcal{W}^{\mathsf{rel}})$. As a result, each original variable in $\mathcal{G}_{\Delta^{\mathsf{rel}}}$ will be detached from its copies, thus ignoring the dependencies encoded by the edges $\mathcal{E}_d$ that were marked to be removed. Algorithm 3 details this procedure. Note that relaxation "breaks" the augmented formula $\Delta^{\mathsf{aug}}$ into a relaxed part $\Delta^{\mathsf{rel}}$ and a "remaining part" $\Delta^{\mathsf{rem}}$, which contains the equivalence constraints just removed.

**Example 4.2.** *Consider the WMI model $(\Delta, \mathcal{W})$ of Example 2.2. Its augmented formula $\Delta^{\mathsf{aug}}$ obtained by applying Algorithm 2 for edges $\mathcal{E}_d = \{X - Z\}$ to be removed (orange), and its relaxed formula $\Delta^{\mathsf{rel}}$ and remaining formula $\Delta^{\mathsf{rem}}$ obtained by Algorithm 3 have their primal graphs shown on the left, center and right below respectively. The detailed WMI models for each are shown in Appendix A.*

**Which edges to relax?** After relaxing enough constraints, we can obtain a WMI problem amenable to exact inference, for example, one whose primal graph $\mathcal{G}_{\Delta^{\mathsf{rel}}}$ has treewidth one and logarithmic




diameter. Running an exact WMI solver on such a problem would already deliver a cheap way to perform approximate inference. However, the quality of such an approximation can be greatly improved if we compensate for the relaxed constraints. We will discuss this in the next section.

A question remains: *how to select the set of edges $\mathcal{E}_d$ to relax?* Note that the more edges we remove from $\Delta$, the easier it is to perform inference on $\Delta^{\mathsf{rel}}$ given fewer dependencies, but the lower the approximation quality, and the harder to compensate for them all, since it would differ from the augmented model more, and meanwhile from the original model as Proposition 4.1 indicates. For example, removing all edges in $\mathcal{G}_\Delta$ will yield a fully disconnected $\mathcal{G}_{\Delta^{\mathsf{rel}}}$ where performing exact inference on each component is going to be embarrassingly parallelizable. This would correpond to perform a loopy version of the MP-WMI algorithm. Analogous to its discrete counterpart, loopy belief propagation, it would be susceptible to poor converge rates [31, 14]. Therefore we propose a simple strategy for selecting the edges to be removed, which is to retrieve a spanning tree of the original primal graph. In Section 6 we demonstrate its practical effectiveness on a range of inference problems of increasing complexity. Devising and evaluating alternative relaxing strategies is an interesting topic for future work.

### 4.2 Compensation

The aim of the compensation phase is to recover the relaxed equivalence constraints and hence, make the distribution $\mathsf{Pr}_{\Delta^{\mathsf{rel}}}$ better approximate $\mathsf{Pr}_{\Delta^{\mathsf{aug}}}$ and thus better approximate $\mathsf{Pr}_\Delta$ as Proposition 4.1 suggests. In order to do so, we introduce new literals, named *compensating literals*, to the variables and their copies in the relaxed formula $\Delta^{\mathsf{rel}}$ and equip them with parameterized weights, named *compensating weights*, and further we optimize them in order to synchronize the variable marginals among a copied variable and its copies.

For each variable $X_i = X_i^0$ and its $C_i$ copies $X_i^1, \ldots, X_i^{C_i}$ in formula $\Delta^{\mathsf{rel}}$, we generate $K$ different univariate literals of the form $\ell_{i,k}^c : (X_i^{(c)} \leq \sigma_{i,k} \cdot \tau_{i,k})$ for $k = 1, \ldots, K$ and $c = 0, 1, \ldots, C_i$ where each $\sigma_{i,k}$ and $\tau_{i,k}$ are respectively drawn at uniform from $\{+1, -1\}$ and the support of $X_i$ as encoded in $\Delta_i^{\mathsf{rel}}$. Note that the $\sigma_{i,k}, \tau_{i,k}$ are *shared* across all the copies. Algorithm 4 in Appendix C summarizes this procedure. Each compensating literal $\ell_{i,k}^c$ is therefore responsible for a portion of the support of the marginal distribution of $X_i^c$, and also for the (unnormalized) marginal density of $X_i^c$ by equipping it with a parameterized weight $w_{\ell_{i,k}^c}$.

To retain tractable inference, the parametric function family chosen for each $w_{\ell_{i,k}^c}$ should satisfy the TWCs as discussed in section 3. Striving for simplicity, we employ constant weights of the form $w_{\ell_{i,k}^c} := \exp(\theta_{i,k}^c)$. Therefore, our induced marginal density takes the form of a piecewise constant approximation. As such, by increasing the number of compensating literals $K$ one could obtain a finer approximation, however at the price of introducing more parameters to optimize for. We empirically investigate the effect of increasing $K$ in our experiments in section 6.

### 4.3 Iterative integration

Instead of matching marginal density functions we settle for the weaker condition of *matching the marginal probabilities of the newly introduced compensating literals*. This in turn can be stated by the following set of equivalence constraints for each variable $X_i$:

$$\mathsf{Pr}_{\Delta^{\mathsf{rem}}}\Big(\bigwedge\nolimits_{c=0}^{C_i} \ell_{k,i}^c\Big) = \mathsf{Pr}_{\Delta^{\mathsf{rel}}}\big(\ell_{k,i}^0\big) = \mathsf{Pr}_{\Delta^{\mathsf{rel}}}\big(\ell_{k,i}^1\big) = \cdots = \mathsf{Pr}_{\Delta^{\mathsf{rel}}}\big(\ell_{k,i}^{C_i}\big), \;\; \textit{for} \;\; k = 1, \cdots, K. \tag{2}$$

where the first term $\mathsf{Pr}_{\Delta^{\mathsf{rem}}}\big(\bigwedge_{j=0}^{C_i} \ell_{k,i}^c\big)$ is the probability of the compensating literals in the remaining WMI model $(\Delta^{\mathsf{rem}}, \mathcal{W}^{\mathsf{rem}})$ and $\mathsf{Pr}_{\Delta^{\mathsf{rel}}}\big(\ell_{k,i}^c\big)$ are the probabilities of compensating literals in the relaxed formula $\Delta^{\mathsf{rel}}$. Intuitively, for a single equivalence constraint that has been relaxed, there exists a set of

**Algorithm 1** ReCoIn $(\Delta, \mathcal{W}, K)$

**Input:** a WMI model $(\Delta, \mathcal{W})$, $K$ number of compensating literals
**Output:** $(\Delta^{\mathsf{rel}}, \mathcal{W}^{\mathsf{rel}})$: a relaxed and compensated WMI model

$\mathcal{E}_d \leftarrow \mathsf{initStrategy}(\Delta, \mathcal{W})$ ▷ Select edges to remove
$\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}}, \mathcal{L} \leftarrow \mathsf{augmentModel}(\Delta, \mathcal{W}, \mathcal{E}_d)$
$(\Delta^{\mathsf{rel}}, \mathcal{W}^{\mathsf{rel}}), (\Delta^{\mathsf{rem}}, \mathcal{W}^{\mathsf{rem}}) \leftarrow \mathsf{relaxModel}(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}}, \mathcal{L})$
$\Delta^{\mathsf{rel}}, \mathcal{W}^{\mathsf{rel}} \leftarrow \mathsf{addingCompensations}(\Delta^{\mathsf{rel}}, \mathcal{W}^{\mathsf{rel}}, \mathcal{L}, K)$
**while** *not* converged **do**
  **for** $X_i \in \mathsf{copiedNodes}(\Delta^{\mathsf{rel}})$ **do**
    **for** $k = 1, \dots, K$ **do**
      $r^k \leftarrow \mathsf{WMI}(\Delta^{\mathsf{rem}}, \mathcal{W}^{\mathsf{rem}}) \,/\, \mathsf{WMI}(\Delta^{\mathsf{rem}} \wedge \bigwedge_{c=0}^{C_i} \ell^c_{k,i}, \mathcal{W}^{\mathsf{rem}}) - 1$
      **for** $c = 0, 1, \dots, C_i$ **do**
        $\theta^{c,(t+1)}_{k,i} \leftarrow \log(r^k \alpha_{k,\sigma(c)}) - \log(1 - \alpha_{k,\sigma(c)}) - \sum_{c' \neq c} \theta^{c,(t)}_{k,i}$
**Return** $(\Delta^{\mathsf{rel}}, \mathcal{W}^{\mathsf{rel}})$

parameters $\theta$ for the compensating weights that exactly match the probabilities in Equation 2 and hence guarantee exact marginal recovery [14]. The next theorem better formalizes it.

**Theorem 4.3.** *Suppose that a relaxed model $(\Delta^{\mathsf{rel}}, \mathcal{W}^{\mathsf{rel}})$ and a remaining model $(\Delta^{\mathsf{rem}}, \mathcal{W}^{\mathsf{rem}})$ are obtained by relaxing a single equivalence constraint $(X_i = X_i^c)$ from an augmented model $\Delta^{\mathsf{aug}}$, and that the primal graph of $\Delta^{\mathsf{rel}}$ is split into two disconnected components by the relaxation. Let $(\ell_{i,k}, \ell^c_{i,k})$ for $k = 1, \dots, K$ be the $K$ pairs of compensating literals introduced, and $\theta_{k,i}, \theta^c_{k,i}$, for $k = 1, \dots, K$, be the parameters attached to the compensating weights. Then Equation 2 holds when the compensating weight parameters satisfy the following equalities.*

$$\theta_{k,i} = \log \frac{r^k \alpha_{k,c}}{1 - \alpha_{k,c}} - \theta^c_{k,i}, \qquad \theta^c_{k,i} = \log \frac{r^k \alpha_k}{1 - \alpha_k} - \theta_{k,i} \quad \textit{for } k = 1, \dots, K \tag{3}$$

*where*

$$r^k = \frac{\mathsf{WMI}(\Delta^{\mathsf{rem}} \wedge \neg \ell_{k,i} \wedge \neg \ell^c_{k,i}, \mathcal{W}^{\mathsf{rem}})}{\mathsf{WMI}(\Delta^{\mathsf{rem}} \wedge \ell_{k,i} \wedge \ell^c_{k,i}, \mathcal{W}^{\mathsf{rem}})}, \;\; \alpha_k = \mathsf{Pr}_{\Delta^{\mathsf{rel}}}(\ell_{i,k}), \; \alpha_{k,c} = \mathsf{Pr}_{\Delta^{\mathsf{rel}}}(\ell^c_{i,k}), \;\; \textit{for } k = 1, \dots, K. \tag{4}$$

Theorem 4.3 suggests an iterative optimization scheme to find the fixed point solutions for all the compensating parameters introduced to compensate multiple relaxed equivalence constraints. Specifically, starting from a random initialization of the parameters of the compensating weights,[3] at each iteration $t + 1$, we can update each parameter $\theta^{c,(t+1)}_{k,i}$ as

$$\theta^{c,(t+1)}_{k,i} \leftarrow \log(r^k \alpha_{k,\pi(c)}) - \log(1 - \alpha_{k,\pi(c)}) - \sum\nolimits_{c' \neq c} \theta^{c',(t)}_{k,i}, \tag{5}$$

where $\pi$ is a permutation over the copies and each $\alpha_{k,\pi(c)}$ is computed as the probability of $\ell^{\pi(c)}_{k,i}$ according to the relaxed model.

Therefore, at each iteration $t$, we need to solve $2K$ integration problems for computing the $r^k$ terms and $C_i \cdot K$ integrations for $\mathsf{Pr}_{\Delta^{\mathsf{rel}}}(\ell^{\pi(c)}_{k,i})$ for each pair of variable and its copies. While in principle we could use any exact WMI solver to solve these problems, we adopt MP-WMI [42] because it is the fastest solver yet for tree-shaped and bounded diameter problems, and even more importantly, it allows to *amortize inference across queries*. That is, we can compute all the $C_i \cdot K$ literal probabilities in a single message-passing step with it.

From this perspective, ReCoIn generates a sequence of induced distributions $\mathsf{Pr}^{(1)}_{\Delta^{\mathsf{rel}}}, \dots, \mathsf{Pr}^{(2)}_{\Delta^{\mathsf{rel}}}, \mathsf{Pr}^{(t)}_{\Delta^{\mathsf{rel}}}$, that should converge to a fixed-point distribution. In practice to check for convergence, one can monitor the quality of the literal probability approximations and stop when a threshold $\epsilon$ is met before

[3] Following Choi and Darwiche [16], we initialize all parameters to 1.

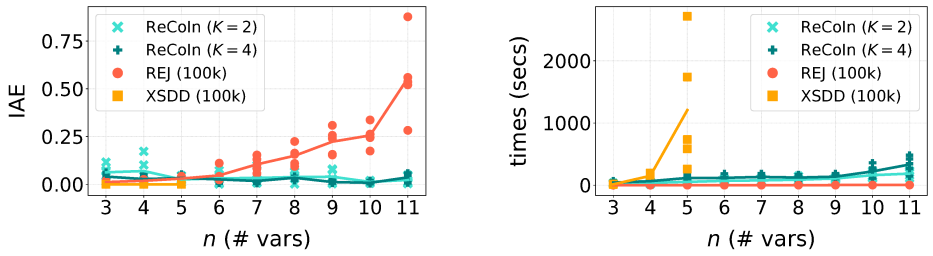



Figure 1: Average integrated absolute errors (left) and times in seconds (right) for 5 problems of increasing size ($n$, x-axis) for ReCoIn and competitors. Number of compensating literals (2-4) or samples used are in parentheses. Mean values per problem size are connected by a line.

a certain number of iterations are done. We choose the threshold to be the maximum $L$-$\infty$ norm of compensation literal probability differences. To ease convergence, we apply *dampening*, that is, we smooth each parameter update at iteration $t+1$ by a factor $\lambda > 0$: $\theta_{k,i}^{c,(t+1)} \leftarrow (1-\lambda)\cdot\theta_{k,i}^{c,(t+1)} + \lambda\cdot\theta_{k,i}^{c,(t+1)}$. This completes the steps in our ReCoIn solver. Algorithm 1 recaps them.

## 5 Related Work

The RCR framework has been particularized for approximating marginals [14, 16, 18], partition functions [17], and for maximization [15] or lifted inference scenarios [37], but always for discrete variables. ReCoIn is the first extension to hybrid domains with SMT($\mathcal{LRA}$) algebraic constraints.

Among the exact WMI solvers, the majority ignores the problem structure to be as general-purpose as possible [8, 34, 35, 29]. However, by doing so they are unable to scale beyond tens of variables in practice. Conversely, recent efficient alternatives such as SMI [41] and MP-WMI [42] can greatly scale but only on WMI problems amenable to tractable inference (cf. Section 3). We leverage the strengths of the latter to efficiently solve iterative integration problems in ReCoIn.

So far, most approximate WMI solvers rely on sampling, and as such inherit all the classical issues of Monte Carlo approaches like poor scalability and convergence [19]. Among these, SAMPO [43] employs Gibbs sampling but does not support generic polynomial weights. A very recent alternative is a fully polynomial randomized approximation scheme [1]. However, it can only operate on DNF SMT formulas, and it is not applicable to our CNF representation as a conversion into DNF can blow up the problem size. Other MCMC variants [3, 2, 4] operating with algebraic constraints, while more effective, cannot be readily used for WMI inference problems. The only alternative to sampling schemes is the hashing-based WMI algorithm [9] which is known to perform poorly on non-trivial problems due the hardness of calibrating the *tilt* [10].

In the next section we compare against the fastest baseline available, the rejection sampler implemented in the pywmi library [30] and a more advanced variant of rejection sampling that greatly increases the acceptance rate of the rejection sampler by compiling an SMT formula into an XSDD structure [44].

## 6 Experiments

We aim to answer the following questions: **(Q1)** how fast and scalable is ReCoIn?, **(Q2)** how accurate are its approximations?, **(Q3)** what is the effect of increasing the number of compensating literals $K$?

We generate WMI problems whose primal graphs are random Watts-Strogatz graphs [39] with increasing size $n = 1, \ldots, 11$, with two additional neighbor connections and probability of rewiring 0.5, to which we attach randomly generated clauses of length 2 and piecewise constant densities. For each setting we generate 5 independent problems.

We run ReCoIn for up to 20 iterations, employing a dampening coefficient $\lambda = 0.5$ in two settings that differ by the number of compensating literals $K = 2, 4$. We compare it against the fastest sampling scheme available, the rejection sampler (REJ) implemented in [30] and the hybrid solver XSDD(Sampling) [44] that employs sophisticated knowledge-compilation [21] techniques [29] to guide sampling. For both REJ and XSDD we employ 100 thousand samples per query.

To compare the quality of approximations for a problem, we compute for a model $\mathcal{M}$ the mean integral absolute error (IAE) as $\frac{1}{|\mathbf{X}|}\sum_{i=1}^{|\mathbf{X}|}\sum_{j=1}^{B}|\mathsf{Pr}_{\mathsf{G}}(X_i \in b_j) - \mathsf{Pr}_{\mathsf{M}}(X_i \in b_j)|$ where we partition the support for each marginal $i = 1, \ldots, n$ into $B$ equal-widths bins $b_j$ for $j = 1, \ldots, B$ and compare the probability $\mathsf{Pr}_{\mathsf{M}}$ according to model $\mathsf{M}$ against the ground truth $\mathsf{Pr}_{\mathsf{G}}$, which we compute using PA [34]. We employ PA as it is so far the most reliable general-purpose exact WMI solver [42]. Note that as such REJ and XSDD are bounded to solve $|\mathbf{X}| \cdot B$ independent WMI problems, while ReCoIn can naturally amortize $|\mathbf{X}| \cdot B$ queries after a single run of optimization (cf. Section 4.3). We impose a timeout of 1 hour.

Figure 1 reports the IAEs and running times (in seconds) for all problems, settings and competitors. Concerning **Q1** and **Q2**, ReCoIn is the best performer overall. The naive sampling strategy in REJ, while being the fastest as expected, cannot exploit the structure in the problem and clearly suffers from the curse of dimensionality. Conversely, XSDD can deliver accurate approximations thanks to compiling the problem structure, but on highly loopy graphs compilation cannot scale beyond $n = 5$. On the other hand, ReCoIn gracefully scales to larger problem sizes and multiple queries, and delivers very low IAE scores that are close to the best by XSDD on small problem sizes. Note that while ReCoIn can solve much larger problems within our timeout, we could not retrieve a ground truth for them with PA in reasonable time (more than 24 hours per problem).

Concerning **Q3**, more compensating literals ($K = 4$) are achieving marginally lower IAEs at the expense of linearly increasing running times. Exploring the time-accuracy trade-off by increasing $K$ or employing different relaxation strategies is an interesting avenue to investigate in the future. All in all, this empirical evidence candidates ReCoIn as one of the best general-purpose approximate WMI solvers in the current landscape of WMI solvers.

## 7 Conclusions

In this work we advanced the WMI framework by tracing the theoretical requirements for tractable WMI inference with the highest precision so far. We introduced ReCoIn as the first solver that by exploiting our tractability insights can reliably scale approximate inference on general WMI problems. We believe these two contributions can help strengthen our theoretical understanding on the challenges and guarantees around approximate hybrid probabilistic inference and at the same time propel the construction of more efficient and scalable WMI solvers.


## Acknowledgement

The authors would like to thank Arthur Choi for several insightful discussions about the RCR framework. This work is partially supported by NSF grants #IIS-1943641, #IIS-1633857,#CCF-1837129, DARPA grant #N66001-17-2-4032, a Sloan Fellowship, Intel, and Facebook. This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. [694980] SYNTH: Synthesising Inductive Data Models).


## Broader Impact

Our contributions in this work can be filed under the label of basic research in probabilistic inference. As a work of basic research it might have a very broad impact. Therefore it is hard to imagine specific negative outcomes at this stage. Concerning benefits, on the other hand, our complexity results will help the community working on probabilistic inference on hybrid domain at large as they lay the foundation for more theoretical research. On the other hand, our general-purpose approximate WMI inference scheme could be particularized by other researchers to fit specific application scenarios. It is hard to foresee or restrict the range of these possible applications. We note that WMI and SMT technologies have been previously used in probabilistic programming and program verification, two very vast fields on their own. Lastly, we are focusing on and advancing inference per se, therefore there is no specific learning phase, or data involved. Our solver is going to perform inference over the distribution induced over an arbitrary SMT theory given as input, if such a theory encodes bias in some form, this bias will clearly be reflected in the probabilistic queries the users are going to ask.

## References


[1] R. Abboud, I. I. Ceylan, and R. Dimitrov. On the approximability of weighted model integration on DNF structures. *arXiv preprint arXiv:2002.06726*, 2020.

[2] H. M. Afshar and J. Domke. Reflection, refraction, and hamiltonian monte carlo. In *Advances in neural information processing systems*, pages 3007–3015, 2015.

[3] H. M. Afshar, S. Sanner, and E. Abbasnejad. Linear-time gibbs sampling in piecewise graphical models. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[4] H. M. Afshar, S. Sanner, and C. Webers. Closed-form gibbs sampling for graphical models with algebraic constraints. In *AAAI*, 2016.

[5] V. Baldoni, N. Berline, J. De Loera, M. Köppe, and M. Vergne. How to integrate a polynomial over a simplex. *Mathematics of Computation*, 80(273):297–325, 2011.

[6] C. Barrett and C. Tinelli. Satisfiability modulo theories. In *Handbook of Model Checking*, pages 305–343. Springer, 2018.

[7] C. Barrett, L. de Moura, S. Ranise, A. Stump, and C. Tinelli. The SMT-LIB initiative and the rise of SMT. In *Proceedings of the 6th international conference on Hardware and software: verification and testing*, pages 3–3. Springer-Verlag, 2010.

[8] V. Belle, A. Passerini, and G. Van den Broeck. Probabilistic inference in hybrid domains by weighted model integration. In *Proceedings of IJCAI*, pages 2770–2776, 2015.

[9] V. Belle, G. Van den Broeck, and A. Passerini. Hashing-based approximate probabilistic inference in hybrid domains. In *UAI*, pages 141–150, 2015.

[10] S. Chakraborty, D. J. Fremont, K. S. Meel, S. A. Seshia, and M. Y. Vardi. Distribution-aware sampling and weighted model counting for sat. In *Proceedings of AAAI*, 2014.

[11] M. Chavira and A. Darwiche. On probabilistic inference by weighted model counting. 2008.

[12] Q. Cheng, J. Hill, and D. Wan. Counting value sets: algorithm and complexity. *The Open Book Series*, 1(1):235–248, 2013.

[13] D. Chistikov, R. Dimitrova, and R. Majumdar. Approximate counting in smt and value estimation for probabilistic programs. *Acta Informatica*, 54(8):729–764, 2017.

[14] A. Choi and A. Darwiche. An edge deletion semantics for belief propagation and its practical impact on approximation quality. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1107. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[15] A. Choi and A. Darwiche. Relax then compensate: On max-product belief propagation and more. In *Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS)*, pages 351–359, 2009.

[16] A. Choi and A. Darwiche. Relax, compensate and then recover. In *JSAI International Symposium on Artificial Intelligence*, pages 167–180. Springer, 2010.

[17] A. Choi and A. Darwiche. Approximating the partition function by deleting and then correcting for model edges. *arXiv preprint arXiv:1206.3241*, 2012.

[18] A. Choi, H. Chan, and A. Darwiche. On bayesian network approximation by edge deletion. *arXiv preprint arXiv:1207.1370*, 2012.

[19] M. K. Cowles and B. P. Carlin. Markov chain monte carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91(434):883–904, 1996.

[20] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki. Incremental task and motion planning: A constraint-based approach. In *Robotics: Science and systems*, volume 12, page 00052. Ann Arbor, MI, USA, 2016.

[21] A. Darwiche and P. Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.

[22] R. Dechter and R. Mateescu. And/or search spaces for graphical models. *Artificial intelligence*, 171(2-3):73–106, 2007.

[23] C. Dehnert, S. Junges, J.-P. Katoen, and M. Volk. A storm is coming: A modern probabilistic model checker. In *International Conference on Computer Aided Verification*, pages 592–600. Springer, 2017.

[24] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[26] C. Haase and S. Kiefer. The complexity of the kth largest subset problem and related problems. *Information Processing Letters*, 116(2):111–115, 2016.

[27] D. Heckerman and D. Geiger. Learning Bayesian networks: a unification for discrete and gaussian domains. In *Proceedings of UAI*, pages 274–284, 1995.

[28] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[29] S. Kolb, M. Mladenov, S. Sanner, V. Belle, and K. Kersting. Efficient symbolic integration for probabilistic inference. In *IJCAI*, pages 5031–5037, 2018.

[30] S. Kolb, P. Morettin, P. Zuidberg Dos Martires, F. Sommavilla, A. Passerini, R. Sebastiani, and L. De Raedt. The pywmi framework and toolbox for probabilistic inference using weighted model integration. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI*, pages 6530–6532, 7 2019. doi: 10.24963/ijcai.2019/946.

[31] D. Koller and N. Friedman. Probabilistic graphical models. 2009.

[32] S. L. Lauritzen and N. Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The annals of Statistics*, pages 31–57, 1989.

[33] A. Molina, A. Vergari, N. Di Mauro, S. Natarajan, F. Esposito, and K. Kersting. Mixed sum-product networks: A deep architecture for hybrid domains. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[34] P. Morettin, A. Passerini, and R. Sebastiani. Efficient weighted model integration via SMT-based predicate abstraction. In *Proceedings of IJCAI*, pages 720–728, 2017.

[35] P. Morettin, A. Passerini, and R. Sebastiani. Advanced smt techniques for weighted model integration. *Artificial Intelligence*, 275:1–27, 2019.

[36] D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1–2):273–302, 1996.

[37] G. Van den Broeck, A. Choi, and A. Darwiche. Lifted relax, compensate and then recover: From approximate to exact lifted probabilistic inference. In *Proceedings of UAI*, pages 131–141, 2012.

[38] A. Vergari, A. Molina, R. Peharz, Z. Ghahramani, K. Kersting, and I. Valera. Automatic bayesian density analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5207–5215, 2019.

[39] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393 (6684):440, 1998.

[40] E. Yang, Y. Baker, P. Ravikumar, G. Allen, and Z. Liu. Mixed graphical models via exponential families. In *Artificial Intelligence and Statistics*, pages 1042–1050, 2014.

[41] Z. Zeng and G. Van den Broeck. Efficient search-based weighted model integration. *Proceedings of UAI*, 2019.

[42] Z. Zeng, P. Morettin, F. Yan, A. Vergari, and G. V. d. Broeck. Scaling up hybrid probabilistic inference with logical and arithmetic constraints via message passing. In *Proceedings of the International Conference of Machine Learning (ICML)*, 2020.

[43] P. M. Zuidberg Dos Martires, A. Dries, and L. De Raedt. Exact and approximate weighted model integration with probability density functions using knowledge compilation. In *Proceedings of the 30th Conference on Artificial Intelligence*. AAAI Press, 2019.

[44] P. M. Zuidberg Dos Martires, S. Kolb, and L. De Raedt. How to exploit structure while solving weighted model integration problems. In *Proceedings of UAI*. AUAI, 2019.

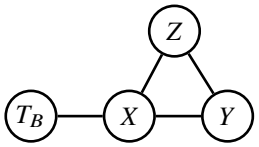
# A Examples

## A.1 Reduction to WMI models on continuous variables only

In this section, we show one example of the polytime reduction from a WMI model with continuos and discrete ones into one over continuous variables only, as introduced in [41].

**Example A.1** (Reduction From WMI to $\mathsf{WMI}_{\mathbb{R}}$)**.** *Consider the WMI model $(\Delta, \mathcal{W})$ where $\Delta$ is the SMT formula over continuous variables $X, Y, Z$ and Boolean variable $B$ as introduced in Example 2.2 with the per-literal weights $\mathcal{W}$ as introduced in Example 2.5. Then the WMI model $(\Delta', \mathcal{W}')$ over continuous variables only $X, Y, Z, T_B$, where $T_B$ is a freshly introduced continuous variable, obtained by the reduction of Zeng and Van den Broeck [41] is shown below.*

$$\Delta' = \begin{cases} 0 \leq X \leq 2 \wedge 1 \leq Y \leq 2 \wedge 0 \leq Z \leq 2 \\ X \geq 1 \vee (-1 \leq T_B \leq 1) \\ X + Y \leq 3 \wedge X + Z \geq 2 \wedge Y + Z \leq 3 \end{cases}$$




*where $\mathcal{W}' = \{w_{\ell_1}(T_B) := 2;\ w_{\ell_2}(x) := x^2;\ w_{\ell_3}(y, z) := 2yz;\ w_{\ell_4}(x, y) := x + y\}$ where $\ell_1 := 0 \leq T_B \leq 1$, $\ell_2 := x \geq 1$, $\ell_3 := y + z \leq 3$, $\ell_4 := x + y \leq 3$ and all the weights associated to other literals are constantly 1 except $\neg\ell_2$ which is 0.*

*Note that the primal graph $\mathcal{G}_{\Delta'}$ (above, right) is isomorphic to the primal graph $\mathcal{G}_\Delta$ and that the weighted model integral of model $(\Delta', \mathcal{W}')$ is left unchanged:*

$$\mathsf{WMI}(\Delta', \mathcal{W}'; X, Y, Z, T_B) = \int_{-1}^{0} dt_B \int_{1}^{2} dx \int_{1}^{-x+3} dy \int_{-x+2}^{-y+3} x^2 \cdot 1 \cdot (x + y) \cdot 2yz\, dz +$$
$$+ \int_{0}^{1} dt_B \int_{1}^{2} dx \int_{1}^{-x+3} dy \int_{-x+2}^{-y+3} x^2 \cdot 2 \cdot (x + y) \cdot 2yz\, dz = \frac{11173}{480} = \mathsf{WMI}(\Delta, \mathcal{W}; X, Y, Z, B).$$

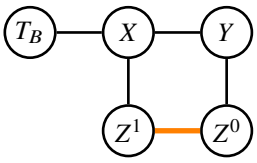
*then we will denote the integrands as $u_1(x, y, z) = x^2 \cdot 1 \cdot (x + y) \cdot 2yz$, $u_2(x, y, z) = x^2 \cdot 2 \cdot (x + y) \cdot 2yz$.*

## A.2 ReCoIn steps: from augmentation to relaxation

Here we complete Example 4.2 by providing the weight functions associated to the WMI models ReCoIn operates on.

**Example A.2** (Augmentation)**.** *Consider the WMI model $(\Delta', \mathcal{W}')$ over continuous variables $X, Y, Z, T_B$ as introduced in Example A.1. Given the edges to remove $\mathcal{E}_d = \{X - Z\}$, the augmented WMI model $(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}})$ over variables $X, Y, Z = Z^0, Z^1, T_B$ as obtained from Algorithm 2 is represented below.*

$$\Delta^{\mathsf{aug}} = \begin{cases} 0 \leq X \leq 2 \wedge 1 \leq Y \leq 2 \\ 0 \leq Z^0 \leq 2 \wedge 0 \leq Z^1 \leq 2 \\ -1 \leq T_B \leq 1 \\ X \geq 1 \vee T_B > 0 \\ X + Y \leq 3 \wedge X + Z^1 \geq 2 \wedge Y + Z^0 \leq 3 \\ Z^0 = Z^1 \end{cases}$$




*and $\mathcal{W}^{\mathsf{aug}} = \{w_{\ell_1}(T_B) := 2;\ w_{\ell_2}(x) := x^2;\ w_{\ell_3}(y, z^0) := 2yz^0;\ w_{\ell_4}(x, y) := x + y;\ w_{\ell_5}(z^0, z^1) := \delta(z^0, z^1)\}$ where $\ell_1 := 0 \leq T_B$, $\ell_2 := x \geq 1$, $\ell_3 := y + z^0 \leq 3$, $\ell_4 := x + y \leq 3$, $\ell_5 := Z^0 = Z^1$ and all the weights associated to other literals are constantly 1 except $\neg\ell_2$ which is 0.*

*Note that the weighted model integral of model $(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}})$ is unchanged as below:*

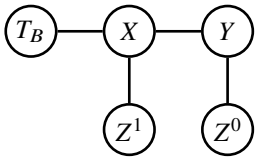
$$\mathsf{WMI}(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}}; X, Y, Z^0, Z^1, T_B) =$$
$$= \int_{-1}^{0} dt_B \int_{1}^{2} dx \int_{1}^{-x+3} dy \int_{0}^{-y+3} \int_{-x+2}^{1} x^2 \cdot (2 + 1) \cdot (x + y) \cdot 2yz^0 \delta(z_0 - z_1) dz^1 dz^0$$
$$= \int_{1}^{2} dx \int_{1}^{-x+3} dy \int_{-x+2}^{-y+3} x^2 \cdot (2 + 1) \cdot (x + y) \cdot 2yz^0\, dz^0$$

$$= \frac{11173}{480} = \mathsf{WMI}(\Delta', \mathcal{W}'; X, Y, Z, T_B) = \mathsf{WMI}(\Delta, \mathcal{W}; X, Y, Z, B).$$

*Further we will show in Proof B.3 that generally the WMI of the augmented model remains unchanged.*

**Example A.3** (Relaxation)**.** *Consider the augmented WMI model* $(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}})$ *over continuous variables* $X, Y, Z^0, Z^1, T_B$ *as introduced in Example A.2. Given the equivalence constraint to remove* $\{Z^0 = Z^1\}$*, the relaxed WMI model* $(\Delta^{\mathsf{rel}}, \mathcal{W}^{\mathsf{rel}})$ *and its remaining part* $(\Delta^{\mathsf{rem}}, \mathcal{W}^{\mathsf{rem}})$ *as obtained from Algorithm 3 are represented below.*

$$\Delta^{\mathsf{rel}} = \begin{cases} 0 \le X \le 2 \wedge 1 \le Y \le 2 \wedge 0 \le Z^0 \le 2 \wedge 0 \le Z^1 \le 2 \\ X \ge 1 \vee (-1 \le T_B \le 1) \\ X + Y \le 3 \wedge X + Z^1 \ge 2 \wedge Y + Z^0 \le 3 \end{cases}$$




$$\Delta^{\mathsf{rem}} = \begin{cases} 0 \le Z^0 \le 2 \wedge 0 \le Z^1 \le 2 \\ Z^0 = Z^1 \end{cases}$$


*and* $\mathcal{W}^{\mathsf{rel}} = \{w_{\ell_1}(T_B) := 2;\ w_{\ell_2}(x) := x^2;\ w_{\ell_3}(y, z^0) := 2yz^0;\ w_{\ell_4}(x, y) := x + y\}$, $\mathcal{W}^{\mathsf{rem}} = \{w_{\ell_5}(z^0, z^1) := \delta(z^0, z^1)\}$*, and all the weights associated to other literals are constantly 1 except* $\neg\ell_2$ *which is 0.*

# B Proofs

## B.1 THEOREM 3.3

*Proof.* We prove our complexity result by reducing a #P-complete variant of the subset sum problem [24] to an MI problem over an SMT($\mathcal{LRA}$) formula $\Delta$ with tree primal graph whose diameter is $n$. This problem is a counting version of subset sum problem saying that given a set of positive integers $S = \{s_1, s_2, \cdots, s_n\}$, and a positive integer $L$, the goal is to count the number of subsets $S' \subseteq S$ such that the sum of all the integers in the subset $S'$ equals to $L$. Notice that our proof can be applied to rational numbers as well and we assume binary representations for numbers.

First, we reduce the counting subset sum problem in polynomial time to a model integration problem by constructing the following SMT($\mathcal{LRA}$) formula $\Delta$ on real variables $\mathbf{X}$ whose primal graph is shown in Figure 2:

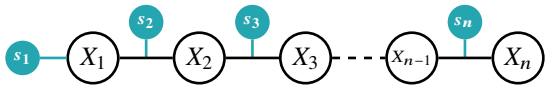



Figure 2: Primal graph $\mathcal{G}_\Delta$ used for the #P-hardness reduction in Theorem 3.3. We construct the corresponding formula $\Delta$ such that $\mathcal{G}_\Delta$ has maximum diameter (it is a chain). We graphically augment graph $\mathcal{G}_\Delta$ by introducing blue nodes to indicate that integers $s_i$ in set $S$ are contained in clauses between two variables.

$$\Delta = \begin{cases} \underbrace{s_1 - \frac{1}{2n} < x_1 < s_1 + \frac{1}{2n}}_{\ell(1,0)} \vee \underbrace{-\frac{1}{2n} < x_1 < \frac{1}{2n}}_{\ell(1,1)} \\ \underbrace{x_{i-1} + s_i - \frac{1}{2n} < x_i < x_{i-1} + s_i + \frac{1}{2n}}_{\ell(i,0)} \ \vee \ \underbrace{x_{i-1} - \frac{1}{2n} < x_i < x_{i-1} + \frac{1}{2n}}_{\ell(i,1)}, \quad i = 2, \cdots n \end{cases}$$

For brevity, we denote the first and the second literal in the $i$-th clause by $\ell(i, 0)$ and $\ell(i, 1)$ respectively as shown above. Also We choose two constants $l = L - \frac{1}{2}$ and $u = L + \frac{1}{2}$.

In the following, we prove that $n^n \mathsf{MI}(\Delta \wedge (l < X_n < u))$ equals to the number of subset $S' \subseteq S$ whose element sum equals to $L$, which indicates that WMI problem whose tree primal graph has diameter $\Theta(n)$ is #P-hard.

Let $\boldsymbol{a}^k = (a_1, a_2, \cdots, a_k)$ be some assignment to Boolean variables $(A_1, A_2, \cdots, A_k)$ with $a_i \in \{0, 1\}$, $i \in [k]$. Given an assignment $\boldsymbol{a}^k$, we define subset sums to be $S(\boldsymbol{a}^k) \triangleq \sum_{i=1}^k a_i s_i$, and formulas $\Delta_{\boldsymbol{a}^k} \triangleq \bigwedge_{i=1}^k \ell(i, a_i)$.

**Claim B.1.** *The model integration for formula $\Delta_{\boldsymbol{a}^k}$ with an given assignment $\boldsymbol{a}^k \in \{0,1\}^k$ is $\mathsf{MI}(\Delta_{\boldsymbol{a}^k}) = (\frac{1}{n})^k$. Moreover, for each variable $X_i$ in $\Delta_{\boldsymbol{a}^k}$, its satisfying assignments consist of the interval $[\sum_{j=1}^i a_j s_j - \frac{i}{2n}, \sum_{j=1}^i a_j s_j + \frac{i}{2n}]$. Specifically, the satisfying assignments for variable $X_n$ in formula $\Delta_{\boldsymbol{a}^n}$ can be denoted by the interval $[S(\boldsymbol{a}^n) - \frac{1}{2}, S(\boldsymbol{a}^n) + \frac{1}{2}]$.*

*Proof.* (Claim B.1) First we prove that $\mathsf{MI}(\Delta_{\boldsymbol{a}^k}) = (\frac{1}{n})^k$. For brevity, denote $a_i s_i$ by $\hat{s_i}$. By definition of model integration and the fact that the integral is absolutely convergent (since we are integrating a constant function, i.e., one, over finite volume regions), we have the following equation.

$$\mathsf{MI}(\Delta_{\boldsymbol{a}^k}) = \int_{(x_1,\cdots,x_k)\models\Delta_{\boldsymbol{a}^k}} 1 \; dx_1 \cdots dx_k = \int_{\hat{s}_1-\frac{1}{2n}}^{\hat{s}_1+\frac{1}{2n}} dx_1 \cdots \int_{x_{k-2}+\hat{s}_{k-1}-\frac{1}{2n}}^{x_{k-2}+\hat{s}_{k-1}+\frac{1}{2n}} dx_{k-1} \int_{x_{k-1}+\hat{s}_k-\frac{1}{2n}}^{x_{k-1}+\hat{s}_k+\frac{1}{2n}} 1 \; dx_k$$

Observe that for the most inner integration over variable $x_k$, the integration result is $\frac{1}{n}$. By doing this iteratively, we have that $\mathsf{MI}(\Delta_{\boldsymbol{a}^k}) = (\frac{1}{n})^k$.

Next we prove that satisfying assignments for variable $X_i$ in formula $\Delta_{\boldsymbol{a}^k}$ is the interval $[\sum_{j=1}^i a_j s_j - \frac{i}{2n}, \sum_{j=1}^i a_j s_j + \frac{i}{2n}]$ by mathematical induction. For $i = 1$, since $X_1$ is in interval $[a_1 s_1 - \frac{1}{2n}, a_1 s_1 + \frac{1}{2n}]$, the statement holds in this case. Suppose that the statement holds for $i = m$, i.e. variable $X_m$ has its satisfying assignments in interval $[\sum_{j=1}^m a_j s_j - \frac{m}{2n}, \sum_{j=1}^m a_j s_j + \frac{m}{2n}]$. Since variable $X_{m+1}$ has its satisfying assignments in interval $[X_m + a_{m+1}s_{m+1} - \frac{1}{2n}, X_m + a_{m+1}s_{m+1} + \frac{1}{2n}]$, then its satisfying assignments consist interval $[\sum_{j=1}^{m+1} a_j s_j - \frac{m+1}{2n}, \sum_{j=1}^{m+1} a_j s_j + \frac{m+1}{2n}]$, that is, the statement also holds for $i = m + 1$. Thus the claim holds. □

The above claim shows how to compute the model integration of formula $\Delta_{\boldsymbol{a}^k}$. We will show in the next claim how to compute the model integration of formula $\Delta_{\boldsymbol{a}^n}$ conjoined with a query $l < X_n < u$.

**Claim B.2.** *For each assignment $\boldsymbol{a}^n \in \{0,1\}^n$, the model integration of formula $\Delta_{\boldsymbol{a}^n} \wedge (l < X_n < u)$ falls into one of the following cases:*

*i) If $S(\boldsymbol{a}^n) < L$ or $S(\boldsymbol{a}^n) > L$, it holds that $\mathsf{MI}(\Delta_{\boldsymbol{a}^n} \wedge (l < X_n < u)) = 0$.*
*ii) If $S(\boldsymbol{a}^n) = L$, it holds that $\mathsf{MI}(\Delta_{\boldsymbol{a}^n} \wedge (l < X_n < u)) = (\frac{1}{n})^n$.*

*Proof.* (Claim B.2) From the previous Claim B.1, it is shown that variable $X_n$ has its satisfying assignments in interval $[S(\boldsymbol{a}^n) - \frac{1}{2}, S(\boldsymbol{a}^n) + \frac{1}{2}]$ in formula $\Delta_{\boldsymbol{a}^n}$ for each $\boldsymbol{a}^n \in \{0,1\}^n$. If $S(\boldsymbol{a}^n) < L$, given that $S(\boldsymbol{a}^n)$ is a sum of positive integers, then it holds that $S(\boldsymbol{a}^n) + \frac{1}{2} \leq (L-1) + \frac{1}{2} = L - \frac{1}{2} = l$ and therefore, $\mathsf{MI}(\Delta_{\boldsymbol{a}^n} \wedge (l < X_n < u)) = 0$; similarly, if $S(\boldsymbol{a}^n) > L$, then it holds that $S(\boldsymbol{a}^n) - \frac{1}{2} \geq u$ and therefore, $\mathsf{MI}(\Delta_{\boldsymbol{a}^n} \wedge (l < X_n < u)) = 0$. If $S(\boldsymbol{a}^n) = L$, by Claim B.1 we have that the satisfying assignment interval is inside the interval $[l, u]$ and thus it holds that $\mathsf{MI}(\Delta_{\boldsymbol{a}^n} \wedge (l < X_n < u)) = \mathsf{MI}(\Delta_{\boldsymbol{a}^n}) = (\frac{1}{n})^n$. □

In the next claim, we show how to compute the model integration of formula $\Delta$ as well as for formula $\Delta$ conjoined with query $l < X_n < u$ based on the already proven Claim B.1 and Claim B.2.

**Claim B.3.** *The following two equations hold:*

*i) $\mathsf{MI}(\Delta) = \sum_{\boldsymbol{a}^n} \mathsf{MI}(\Delta_{\boldsymbol{a}^n})$.*
*ii) $\mathsf{MI}(\Delta \wedge (l < X_n < u)) = \sum_{\boldsymbol{a}^n} \mathsf{MI}(\Delta_{\boldsymbol{a}^n} \wedge (l < X_n < u))$.*

*Proof.* (Claim B.3) Observe that for each clause in $\Delta$, literals are mutually exclusive since each $s_i$ is a positive integer. Then we have that formulas $\Delta_{\boldsymbol{a}^n}$ are mutually exclusive and meanwhile $\Delta = \bigvee_{\boldsymbol{a}^n} \Delta_{\boldsymbol{a}^n}$. Thus it holds that $\mathsf{MI}(\Delta) = \sum_{\boldsymbol{a}^n} \mathsf{MI}(\Delta_{\boldsymbol{a}^n})$. Similarly, we have formulas $(\Delta_{\boldsymbol{a}^n} \wedge (l < X_n < u))$'s are mutually exclusive and meanwhile $\Delta \wedge (l < X_n < u) = \bigvee_{\boldsymbol{a}^n} \Delta_{\boldsymbol{a}^n} \wedge (l < X_n < u)$. Thus the second equation holds. □

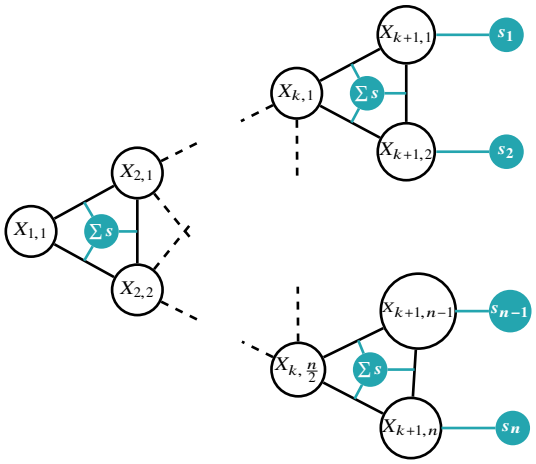



Figure 3: Primal graph used for #P-hardness reduction in Theorem 7. We also put blue nodes to indicate that integer $s_i$'s in set $S$ are contained in some clauses and that model integration over some cliques is the sum of some $s_i$'s.

From the above claims, we can conclude that $\mathsf{MI}(\Delta \wedge (l < X_n < u)) = t(\frac{1}{n})^n$ where $t$ is the number of assignments $\boldsymbol{a}^n$ s.t. $S(\boldsymbol{a}^n) = L$. Notice that for each $\boldsymbol{a}^n \in \{0,1\}^n$, there is a one-to-one correspondance to a subset $S' \subseteq S$ by defining $\boldsymbol{a}^n$ as $a_i = 1$ if and only if $s_i \in S'$; and $S(\boldsymbol{a}^n)$ equals to $L$ if and only if the sum of elements in $S'$is $L$. Therefore $n^n\mathsf{MI}(\Delta \wedge (l < X_n < u))$ equals to the number of subset $S' \subseteq S$ whose element sum equals to $L$. This finishes the proof for the statement that inference in $\mathcal{WMJ}(\boldsymbol{\Omega}, n, 1)$ is #P-hard.

□

### B.2 THEOREM 3.4

*Proof.* Again we prove our complexity result by reducing the #P-complete variant of the subset sum problem [24] to an MI problem over an SMT($\mathcal{LRA}$) formula $\Delta$ with primal graph whose diameter is $\Theta(\log n)$ and treewidth two. In the #P-complete subset sum problem, we are given a set of positive integers $S = \{s_1, s_2, \cdots, s_n\}$, and a positive integer $L$. Notice that our proof can be applied to rational numbers as well and we assume binary representations for numbers. The goal is to count the number of subsets $S' \subseteq S$ such that the sum of all the integers in $S'$ equals $L$.

First, we reduce this problem in polynomial time to a model integration problem with the following SMT($\mathcal{LRA}$) formula $\Delta$ where variables are real and $u$ and $l$ are two constants. Its primal graph is shown in Figure 3. Consider $n = 2^k$, $n, k \in \mathbb{N}$.

$$\Delta = \bigwedge_{i\in[n]} (-\frac{1}{4n} < X_{k+1,i} < \frac{1}{4n} \vee -\frac{1}{4n} + s_i < X_{k+1,i} < \frac{1}{4n} + s_i) \bigwedge \Delta_t$$

$$\text{where } \Delta_t = \bigwedge_{j\in[k], i\in[2^j]} -\frac{1}{4n} + X_{j+1,2i-1} + X_{j+1,2i} < X_{j,i} < \frac{1}{4n} + X_{j+1,2i-1} + X_{j+1,2i}$$

For brevity, we denote all the variables by $\mathbf{X}$ and denote the literal $-\frac{1}{4n} < X_{k+1,i} < \frac{1}{4n}$ by $\ell(i,0)$ and literal $-\frac{1}{4n} + s_i < X_{k+1,i} < \frac{1}{4n} + s_i$ by $\ell(i,1)$ respectively. Also We choose two constants $l = L - \frac{1}{2}$ and $u = L + \frac{1}{2}$. In the following, we prove that $(2n)^{2n-1}\mathsf{MI}(\Delta \wedge (l < X_{1,1} < u))$ equals to the number of subset $S' \subseteq S$ whose element sum equals to $L$, which indicates that model integration problem with primal graph whose diameter is $\Theta(\log n)$ and treewidth two is #P-hard.

Let $\boldsymbol{a}^n = (a_1, a_2, \cdots, a_n) \in \{0,1\}^n$ be some assignment to Boolean variables $(A_1, A_2, \cdots, A_n)$. Given an assignment $\boldsymbol{a}^n$, define the sum as $S(\boldsymbol{a}^n) \triangleq \sum_{i=1}^n a_i s_i$, and formula as $\Delta_{\boldsymbol{a}^n} \triangleq \bigwedge_{i=1}^n \ell(i, a_i) \wedge \Delta_t$.

**Claim B.4.** *The model integration for formula $\Delta_{\boldsymbol{a}^n}$ with given $\boldsymbol{a}^n \in \{0,1\}^n$ is $\mathsf{MI}(\Delta_{\boldsymbol{a}^n}) = (\frac{1}{2n})^{2n-1}$. Moreover, for each variable $X_{j,i}$ in formula $\Delta_{\boldsymbol{a}^n}$, its satisfying assignments consist of the interval $[\sum_l a_l s_l - \frac{2^{k-j+2}-1}{4n}, \sum_l a_l s_l + \frac{2^{k-j+2}-1}{4n}]$ where $l \in \{l \mid X_{k+1,l}$ is a descendant of $X_{j,i}\}$. Specifically, the satisfying assignments for the root variable $X_{1,1}$ can be denoted the interval $[S(\boldsymbol{a}^n) - \frac{2n-1}{4n}, S(\boldsymbol{a}^n) + \frac{2n-1}{4n}] \subset [S(\boldsymbol{a}^n) - \frac{1}{2}, S(\boldsymbol{a}^n) + \frac{1}{2}]$.*

*Proof.* (Claim B.4) First we prove that $\mathsf{MI}(\Delta_{\boldsymbol{a}^n}) = (\frac{1}{2n})^{2n-1}$. For brevity, denote $a_i s_i$ by $\hat{s_i}$. By definition of model integration and the fact that the integral is absolutely convergent (since we are integrating a constant function, i.e., one, over finite volume regions), we have the following equations

$$\begin{aligned}\mathsf{MI}(\Delta_{\boldsymbol{a}^n}) &= \int_{\boldsymbol{x} \models \Delta_{\boldsymbol{a}^n}} 1 \, d\mathbf{X} \\ &= \int_{-\frac{1}{4n}+\hat{s}_n}^{\frac{1}{4n}+\hat{s}_n} dx_{k+1,n} \cdots \int_{-\frac{1}{4n}+\hat{s}_1}^{\frac{1}{4n}+\hat{s}_1} dx_{k+1,1} \int_{-\frac{1}{4n}+x_{k+1,n-1}+x_{k+1,n}}^{\frac{1}{4n}+x_{k+1,n-1}+x_{k+1,n}} dx_{k,2^{k-1}} \cdots \int_{-\frac{1}{4n}+x_{2,1}+x_{2,2}}^{\frac{1}{4n}+x_{2,1}+x_{2,2}} 1 \, dx_{1,1} \, .\end{aligned}$$

Observe that for the most inner integration over variable $x_{1,1}$, the integration result is $\frac{1}{2n}$. By doing this iteratively, we have that $\mathsf{MI}(\Delta_{\boldsymbol{a}^k}) = (\frac{1}{2n})^{2n-1}$ where the $2n-1$ comes from the number of variables.

Then we prove that satisfying assignments for variable $X_{j,i}$ in formula $\Delta_{\boldsymbol{a}^n}$ lie in the interval $[\sum_l a_l s_l - \frac{2^{k-j+2}-1}{4n}, \sum_l a_l s_l + \frac{2^{k-j+2}-1}{4n}]$ where $l \in \{l \mid X_{k+1,l}$ is a descendant of $X_{j,i}\}$ by performing mathematical induction in a bottom-up way.

For $j = 1$, any variable $X_{k+2-j,i}$ with $i \in [2^{k+2-j}]$ has satisfying assignments consisting of the interval $[a_i s_i - \frac{1}{4n}, a_i s_i + \frac{1}{4n}]$. Thus the statement holds for this case.

Suppose that the statement holds for $j = m$, that is, for any $i \in [2^{k+2-m}]$, any variable $X_{k+2-m,i}$ has satisfying assignments consisting interval $[\sum_l a_l s_l - \frac{2^m-1}{4n}, \sum_l a_l s_l + \frac{2^m-1}{4n}]$ where $l \in \{l \mid X_{k+1,l}$ is a descendant of $X_{k+2-m,i}\}$.

Then for $j = m+1$ and any $i \in [2^{k+1-m}]$, the variable $X_{k+1-m,i}$ has two descendants, variable $X_{k+2-m,2i-1}$ and variable $X_{k+2-m,2i}$. Moreover, we have that $-\frac{1}{4n} + X_{k+2-m,2i-1} + X_{k+2-m,2i} < X_{k+1-m,i} < \frac{1}{4n} + X_{k+2-m,2i-1} + X_{k+2-m,2i}$. Then the lower bound of the interval for variable $X_{k+1-m,i}$ is $-\frac{1}{4n} + \sum_l a_l s_l - 2\frac{2^m-1}{4n} = \sum_l a_l s_l - \frac{2^{m+1}-1}{4n}$; similarly the upper bound of the interval is $\sum_l a_l s_l + \frac{2^{m+1}-1}{4n}$, where $l \in \{l \mid X_{k+1,l}$ is a descendant of $X_{k+1-m,i}\}$. That is, the statement also holds for $j = m+1$ which finishes our proof. □

The above claim shows what the model integration of formula $\Delta_{\boldsymbol{a}^k}$ is like. We'll show in the next claim what the model integration of formula $\Delta_{\boldsymbol{a}^n}$ conjoined with a query $l < X_{1,1} < u$ is like.

**Claim B.5.** *For each assignments $\boldsymbol{a}^n \in \{0,1\}^n$, the model integration of $\Delta_{\boldsymbol{a}^n} \wedge (l < X_{1,1} < u)$ falls into one of the following cases:*

*i) If $S(\boldsymbol{a}^n) < L$ or $S(\boldsymbol{a}^n) > L$, then $\mathsf{MI}(\Delta_{\boldsymbol{a}^n} \wedge (l < X_{1,1} < u)) = 0$.*
*ii) If $S(\boldsymbol{a}^n) = L$, then $\mathsf{MI}(\Delta_{\boldsymbol{a}^n} \wedge (l < X_{1,1} < u)) = (\frac{1}{2n})^{2n-1}$.*

*Proof.* (Claim B.5) From previous Claim B.4, it is shown that variable $X_{1,1}$ has its satisfying assignments in the interval $[S(\boldsymbol{a}^n) - \frac{2n-1}{4n}, S(\boldsymbol{a}^n) + \frac{2n-1}{4n}]$ in formula $\Delta_{\boldsymbol{a}^n}$ for each $\boldsymbol{a}^n \in \{0,1\}^n$.

If $S(\boldsymbol{a}^n) < L$, given that $S(\boldsymbol{a}^n)$ is a sum of positive integers, then it holds that $S(\boldsymbol{a}^n) + \frac{1}{2} \leq (L-1) + \frac{2n-1}{4n} < L - \frac{1}{2} = l$ and therefore, $\mathsf{MI}(\Delta_{\boldsymbol{a}^n} \wedge (l < X_{1,1} < u)) = 0$; similarly, if $S(\boldsymbol{a}^n) > L$, then it holds that $S(\boldsymbol{a}^n) - \frac{1}{2} > u$ and therefore, $\mathsf{MI}(\Delta_{\boldsymbol{a}^n} \wedge (l < X_{1,1} < u)) = 0$. If $S(\boldsymbol{a}^n) = L$, then by Claim B.4 we have that the satisfying assignment interval is inside the interval $[l, u]$ and thus it holds that $\mathsf{MI}(\Delta_{\boldsymbol{a}^n} \wedge (l < X_{1,1} < u)) = \mathsf{MI}(\Delta_{\boldsymbol{a}^n}) = (\frac{1}{2n})^{2n-1}$. □

**Claim B.6.** *The following two equations hold:*

*i) $\mathsf{MI}(\Delta) = \sum_{\boldsymbol{a}^n} \mathsf{MI}(\Delta_{\boldsymbol{a}^n})$.*
*ii) $\mathsf{MI}(\Delta \wedge (l < X_{1,1} < u)) = \sum_{\boldsymbol{a}^n} MI(\Delta_{\boldsymbol{a}^n} \wedge (l < X_{1,1} < u))$.*

*Proof.* (Claim B.6) Observe that for each pair of literals $\ell(i,0)$ and $\ell(i,1), i \in [n]$, literals are mutually exclusive since each $s_i$ is a positive integer. Then we have that formulas $\Delta_{\boldsymbol{a}^n}$ are mutually exclusive and meanwhile formula $\Delta = \bigvee_{\boldsymbol{a}^n} \Delta_{\boldsymbol{a}^n}$. Thus it holds that $\mathsf{MI}(\Delta) = \sum_{\boldsymbol{a}^n} \mathsf{MI}(\Delta_{\boldsymbol{a}^n})$. Similarly, we have formulas $(\Delta_{\boldsymbol{a}^n} \wedge (l < X_{1,1} < u))$'s are mutually exclusive and meanwhile $\Delta \wedge (l < X_{1,1} < u) = \bigvee_{\boldsymbol{a}^n} \Delta_{\boldsymbol{a}^n} \wedge (l < X_{1,1} < u)$. Thus the second equation holds. □

From the above claims, we can conclude that $\mathsf{MI}(\Delta \wedge (l < X_{1,1} < u)) = t(\frac{1}{2n})^{2n-1}$ where $t$ is the number of assignments $\boldsymbol{a}^n$ s.t. $S(\boldsymbol{a}^n) = L$. Notice that for each $\boldsymbol{a}^n \in \{0,1\}^n$, there is a one-to-one correspondence to a subset $S' \subseteq S$ by defining $\boldsymbol{a}^n$ as $a_i = 1$ if and only if $s_i \in S'$; and $S(\boldsymbol{a}^n)$ equals to $L$ if and only if the sum of elements in $S'$ is $L$. Therefore $(2n)^{2n-1}\mathsf{MI}(\Delta \wedge (l < X_{1,1} < u))$ equals to the number of subset $S' \subseteq S$ whose element sum equals to $L$. This finishes the proof for the statement that inference in $\mathcal{WMJ}(\boldsymbol{\Omega}, \log(n), 2)$ is #P-hard. □

### B.3 PROPOSITION 4.1

*Proof.* W.l.o.g, consider the case where the augmented WMI model $(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}})$ is obtained by removing an edge $X_i - X_j$ and inducing the dependency $X_i - X_i^c - X_j$ from the original WMI model $(\Delta, \mathcal{W})$ as shown in Algorithm 2.

Instrumentally to the proof, we introduce the concept of *total truth assignments* of an SMT($\mathcal{LRA}$) formula $\Delta$. A total truth assignment $\mu$ is defined as a partitioning of all true literals in $\mathcal{L}$, the set of all literals in formula $\Delta$, into a set of literals $\mu_\top$ interpreted as true for a certain total configurations of the variables in $\Delta$ and and the complementary set $\mu_\bot$ containing the literals interpreted as false. Let $tta(\Delta)$ be the set of all total truth assignments for formula $\Delta$.

Notice that when operating on continuous variables only, the definition of WMI in Equation 1 can be rewritten in terms of the total truth assignments to $\Delta$ as follows:

$$\mathsf{WMI}(\Delta, \mathcal{W}) = \sum_{\mu \in tta(\Delta)} \int [\![\boldsymbol{x} \models \mu]\!] \prod_{\ell \in \mathcal{L}} w(\boldsymbol{x})^{[\![\boldsymbol{x} \models \ell]\!]} d\boldsymbol{x} := \sum_{\mu \in tta(\Delta)} Z_\mu. \tag{6}$$

Before we prove that the WMI remains unchanged for the augmented model, we need the following claim.

**Claim B.7.** *Let $tta(\Delta)$ and $tta(\Delta^{\mathsf{aug}})$ be the set of total truth assignments of formula $\Delta$ and that of formula $\Delta^{\mathsf{aug}}$ respectively. Then there exists a bijection between $tta(\Delta)$ and $tta(\Delta^{\mathsf{aug}})$.*

*Proof.* The proof is done by explicitly constructing a bijection $f : tta(\Delta) \to tta(\Delta^{\mathsf{aug}})$ which maps $\mu \in tta(\Delta)$ to $\mu' \in tta(\Delta^{\mathsf{aug}})$ in the following way:

i) for every $\ell \in \Delta_i$, if $\ell \in \mu_\top$, then $\ell \in \mu'_\top$ and $\ell\{X_i : X_i^c\} \in \mu'_\top$; otherwise $\ell \in \mu'_\bot$ and $\ell\{X_i : X_i^c\} \in \mu'_\bot$.
ii) for every $\ell \in \Delta_{ij}$, if $\ell \in \mu_\top$, then $\ell\{X_i : X_i^c\} \in \mu'_\top$; otherwise $\ell\{X_i : X_i^c\} \in \mu'_\bot$.
iii) for every $\ell \notin \Delta_i$ and $\ell \notin \Delta_{ij}$, if $\ell \in \mu_\top$, then $\ell \in \mu'_\top$; otherwise $\ell \in \mu'_\bot$.
iv) finally, by definition, literal $X_i = X_i^c$ is always in set $\mu'_\top$ (otherwise $\mu'$ would not be a satisfying assignment to formula $\Delta^{\mathsf{aug}}$)

where $\Delta_i$ is the sub-formula containing all the univariate clauses in $\Delta$ referring to $X_i$ only and analogously $\Delta_{ij}$ is the sub-formula containing bivariate clauses in $\Delta$ referring to $X_i$ and $X_j$.

First, note that the function $f$ is well-defined since every literal in formula $\Delta^{\mathsf{aug}}$ is assigned to either set $\mu'_\top$ or set $\mu'_\bot$ by the construction of formula $\Delta^{\mathsf{aug}}$ and this uniquely defines a $\mu' \in tta(\Delta^{\mathsf{aug}})$. Second, by construction, if $f(\mu_1) = f(\mu_2)$ for some $\mu_1, \mu_2 \in tta(\Delta)$, the two total truth assignments $\mu_1$ and $\mu_2$ should have the same set of positive literals as well as the same set of negative literals, which means that $\mu_1 = \mu_2$. Thus, the function $f$ is a one-to-one mapping. Moreover, for each $\mu' \in tta(\Delta^{\mathsf{aug}})$, there exists $\mu \in tta(\Delta)$ obtained by substituting the variable $x'_i$ by $X_i$ and deleting literals in $\Delta_i\{X_i : X_i^c\}$ and literal $X_i = X_i^c$, such that $f(\mu) = \mu'$. That is, the function $f$ is also an onto mapping. Overall, the function $f$ is a bijection between $tta(\Delta)$ and $tta(\Delta^{\mathsf{aug}})$. □

From Equation 6, it follows that to prove that $\mathsf{WMI}(\Delta, \mathcal{W}) = \mathsf{WMI}(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}})$, it suffices to prove that for each $\mu \in tta(\Delta)$, $Z_\mu$, the integration inside summation corresponding to assignment $\mu$, equates

$Z^{\mathsf{aug}}_{f(\mu)}$ inside $\mathsf{WMI}(\Delta^{\mathsf{aug}}$ with function $f$ as defined in Claim B.7. Let $\mathbf{X}_{-i} = \mathbf{X} \setminus \{X_i\}$. Then the set of variables appearing in formula $\Delta^{\mathsf{aug}}$ can be written as $\mathbf{X}_{-i} \cup \{X_i\} \cup \{X^c_i\}$. Let $\Delta^{\mathsf{aug}}_{ij} := \Delta_{ij}\{X_i : X^c_i\}$ and $\Delta^{\mathsf{aug}} := \Delta^{\bar{\mathsf{aug}}} \wedge (X_i = X^c_i)$. We explicitly formulate the integration $Z_\mu$ and $Z^{\mathsf{aug}}_{f(\mu)}$ as follows.

$$
\begin{aligned}
Z_\mu &= \int [\![\boldsymbol{x} \models \mu]\!] \prod_{\ell \in \Delta} w_\ell(\boldsymbol{x})^{[\![\boldsymbol{x} \models \ell]\!]} d\boldsymbol{x} \\
Z^{\mathsf{aug}}_{f(\mu)} &= \int [\![\boldsymbol{x}_{-i}, x_i, x^c_i \models f(\mu)]\!] \prod_{\ell \in \Delta^{\bar{\mathsf{aug}}}} w_\ell(\boldsymbol{x}_{-i}, x_i, x^c_i)^{[\![\boldsymbol{x}_{-i}, x_i, x^c_i \models \ell]\!]} \delta(x_i - x^c_i) dx^c_i \, dx_i d\boldsymbol{x}_{-i} \\
&= \int \prod_{\substack{\ell \in \Delta^{\bar{\mathsf{aug}}} \\ \ell \notin \Delta^{\mathsf{aug}}_{ij}}} w_\ell(\boldsymbol{x}_{-i}, x_i)^{[\![\boldsymbol{x}_{-i}, x_i \models \ell]\!]} \cdot \\
&\qquad \left( \int \prod_{\ell \in \Delta^{\mathsf{aug}}_{ij}} w_\ell(x^c_i, x_j)^{[\![x^c_i, x_j \models \ell]\!]} \delta(x_i - x^c_i) [\![\boldsymbol{x}_{-i}, x_i, x^c_i \models f(\mu)]\!] dx^c_i \right) dx_i d\boldsymbol{x}_{-i}
\end{aligned}
$$

Notice that by the property of Dirac Delta function and the construction of function $f$, it holds that

$$
\int \prod_{\ell \in \Delta^{\mathsf{aug}}_{ij}} w_\ell(x^c_i, x_j)^{[\![x^c_i, x_j \models \ell]\!]} \delta(x_i - x^c_i) [\![\boldsymbol{x}_{-i}, x_i, x^c_i \models f(\mu)]\!] dx^c_i = \prod_{\ell \in \Delta_{ij}} w(x_i, x_j)^{[\![x_i, x_j \models \ell]\!]} [\![\boldsymbol{x} \models \mu]\!]
$$

Therefore, it holds that

$$
Z^{\mathsf{aug}}_{f(\mu)} = \int [\![\boldsymbol{x} \models \mu]\!] \prod_{\substack{\ell \in \Delta^{\bar{\mathsf{aug}}} \\ \ell \notin \Delta^{\mathsf{aug}}_{ij}}} w_\ell(\boldsymbol{x}_{-i}, x_i)^{[\![\boldsymbol{x}_{-i}, x_i \models \ell]\!]} \prod_{\ell \in \Delta_{ij}} w_\ell(x_i, x_j)^{[\![x_i, x_j \models \ell]\!]} d\boldsymbol{x} = Z_\mu
$$

Finally, we have that the WMI of the original model $(\Delta, \mathcal{W})$ equates that of the augmented model $(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}})$ by observing that $\mathsf{WMI}(\Delta, \mathcal{W}) = \sum_\mu Z_\mu = \sum_{f(\mu)} Z^{\mathsf{aug}}_{f(\mu)} = \mathsf{WMI}(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}})$.

Moreover, for any univariate literal $\ell$, it can be shown by similar arguments that $\mathsf{WMI}(\Delta \wedge \ell, \mathcal{W}) = \mathsf{WMI}(\Delta^{\mathsf{aug}} \wedge \ell, \mathcal{W}^{\mathsf{aug}})$. Thus, it holds that $\mathsf{Pr}_\Delta(\ell) = \mathsf{WMI}(\Delta \wedge \ell, \mathcal{W})/\mathsf{WMI}(\Delta, \mathcal{W}) = \mathsf{WMI}(\Delta^{\mathsf{aug}} \wedge \ell, \mathcal{W}^{\mathsf{aug}})/\mathsf{WMI}(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}}) = \mathsf{Pr}_{\Delta^{\mathsf{aug}}}(\ell)$. □

### B.4 THEOREM 4.3

For the remaining WMI model $(\Delta^{\mathsf{rem}}, \mathcal{W}^{\mathsf{rem}})$, it holds that

$$
\begin{aligned}
\mathsf{Pr}_{\Delta^{\mathsf{rem}}}(\ell_{k,i} \wedge \ell^c_{k,i}) &= \frac{\mathsf{WMI}(\Delta^{\mathsf{rem}} \wedge \ell_{k,i} \wedge \ell^c_{k,i}, \mathcal{W}^{\mathsf{rem}})}{\mathsf{WMI}(\Delta^{\mathsf{rem}}, \mathcal{W}^{\mathsf{rem}})} \\
&= \frac{\mathsf{WMI}(\Delta^{\mathsf{rem}} \wedge \ell_{k,i} \wedge \ell^c_{k,i}, \mathcal{W}^{\mathsf{rem}})}{\mathsf{WMI}(\Delta^{\mathsf{rem}} \wedge \ell_{k,i} \wedge \ell^c_{k,i}, \mathcal{W}^{\mathsf{rem}}) + \mathsf{WMI}(\Delta^{\mathsf{rem}} \wedge \neg\ell_{k,i} \wedge \neg\ell^c_{k,i}, \mathcal{W}^{\mathsf{rem}})} \\
&= \frac{\exp(\theta_{k,i} + \theta^c_{k,i})}{r^k + \exp(\theta_{k,i} + \theta^c_{k,i})}
\end{aligned}
$$

By substituting the sum of $\theta_{k,i}$ and $\theta^c_{k,i}$ with the first equality in Equation 3, it holds that $\mathsf{Pr}_{\Delta^{\mathsf{rem}}}(\ell_{k,i} \wedge \ell^c_{k,i}) = \mathsf{Pr}_{\Delta^{\mathsf{rel}}}(\ell_{k,i})$; similarly, by substituting the sum with the second equality, it holds that $\mathsf{Pr}_{\Delta^{\mathsf{rem}}}(\ell_{k,i} \wedge \ell^c_{k,i}) = \mathsf{Pr}_{\Delta^{\mathsf{rel}}}(\ell^c_{k,i})$, which finishes the proof.

# C Algorithms

**Algorithm 2** augmentModel($\Delta, \mathcal{W}, \mathcal{E}_d$)

**Input:** a WMI model with SMT formula $\Delta$ and per-literal weights $\mathcal{W}$ and a set $\mathcal{E}_d$ of edges to be deleted

**Output:** augmented WMI model $(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}})$ and equivalence constraint set $\mathcal{L}$

$\Delta^{\mathsf{aug}} \leftarrow \mathsf{copy}(\Delta)$
$\mathcal{W}^{\mathsf{aug}} \leftarrow \mathsf{copy}(\mathcal{W})$
$\mathcal{L} \leftarrow \{\}$
**for** edge $X_i - X_j \in \mathcal{E}_d$ **do**
    $X_i^c \leftarrow \mathsf{copy}(X_i)$ ▷ Assume to copy $X_i$
    $\hat{\ell} \leftarrow (X_i = X_i^c)$
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{\hat{\ell}\}$
    $\Delta' \leftarrow \Delta^{\mathsf{aug}} \wedge \hat{\ell}$,
    $w_{\hat{\ell}} := \delta(X_i, X_i^c)$
    $\mathcal{W}^{\mathsf{aug}} \leftarrow \mathcal{W}^{\mathsf{aug}} \cup \{w_{\hat{\ell}}\}$
    **for** clause $\Gamma \in \Delta_{i,j}$ **do** ▷ Rename edges
        $\Gamma' \leftarrow \Gamma\{X_i : X_i^c\}$
        $\Delta' \leftarrow \Delta'\{\Gamma : \Gamma'\}$
        **for each** literal $\ell \in \Gamma$ **do**
            $\ell' \leftarrow \ell\{X_i : X_i^c\}$
            $w_{\ell'} \leftarrow \mathsf{copy}(w_\ell)$
            $\mathcal{W}^{\mathsf{aug}} \leftarrow \mathcal{W}^{\mathsf{aug}} \cup \{w_{\ell'}\} \setminus \{w_\ell\}$
    **for** clause $\Gamma \in \Delta_i$ **do** ▷ Copy and rename bounding-box literals
        $\Gamma' \leftarrow \mathsf{copy}(\Gamma)$
        $\Delta' \leftarrow \Delta' \wedge \Gamma'\{X_i : X_i^c\}$
    $\Delta^{\mathsf{aug}} \leftarrow \Delta'$
**return** $\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}}, \mathcal{L}$

**Algorithm 3** relaxModel($\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}}, \mathcal{L}$)

**Input:** an augmented WMI model $(\Delta^{\mathsf{aug}}, \mathcal{W}^{\mathsf{aug}})$, $\mathcal{L}$: equivalence constraints to be relaxed
**Output:** a relaxed WMI model $(\Delta^{\mathsf{rel}}, \mathcal{W}^{\mathsf{rel}})$, and its "remaining-part" model $(\Delta^{\mathsf{rem}}, \mathcal{W}^{\mathsf{rem}})$.

$\Delta^{\mathsf{rem}} \leftarrow \top$
$\mathcal{W}^{\mathsf{rem}} \leftarrow \{\}$
$\Delta^{\mathsf{rel}} \leftarrow \mathsf{copy}(\Delta^{\mathsf{aug}})$
$\mathcal{W}^{\mathsf{rel}} \leftarrow \mathsf{copy}(\mathcal{W}^{\mathsf{aug}})$
**for each** $\ell^* : (X_i = X_i^c) \in \mathcal{L}$ **do**
    **for** clause $\Gamma \in \Delta_i$ **do**
        $\Delta^{\mathsf{rem}} \leftarrow \Delta^{\mathsf{rem}} \wedge \Gamma \wedge \Gamma\{X_i : X_i^c\}$
        **for each** literal $\ell \in \Gamma$ **do**
            $\ell' \leftarrow \ell\{X_i : X_i^c\}$
            $w_{\ell'} \leftarrow \mathsf{copy}(w_\ell)$
            $\mathcal{W}^{\mathsf{rel}} \leftarrow \mathcal{W}^{\mathsf{rel}} \cup \{w_{\ell'}\}$
            $\mathcal{W}^{\mathsf{rem}} \leftarrow \mathcal{W}^{\mathsf{rem}} \cup \{w_\ell, w_{\ell'}\}$
    $\Delta^{\mathsf{rel}} \leftarrow \Delta^{\mathsf{rel}}\{\ell^* : \top\}$ ▷ disconnect $X_i$ and copy $X_i^c$
    $\mathcal{W}^{\mathsf{rel}} \leftarrow \mathcal{W}^{\mathsf{rel}} \setminus \{w_{\ell^*}\}$
    $\Delta^{\mathsf{rem}} \leftarrow \Delta^{\mathsf{rem}} \wedge \ell^*$
    $\mathcal{W}^{\mathsf{rem}} \leftarrow \mathcal{W}^{\mathsf{rem}} \cup \{w_{\ell^*}\}$
**return** $(\Delta^{\mathsf{rel}}, \mathcal{W}^{\mathsf{rel}})$, $(\Delta^{\mathsf{rem}}, \mathcal{W}^{\mathsf{rem}})$

**Algorithm 4** addingCompensations($\Delta^{\mathsf{rel}}, \mathcal{W}^{\mathsf{rel}}, \mathcal{L}, K$)

**Input:** a relaxed WMI model $(\Delta^{\mathsf{rel}}, \mathcal{W}^{\mathsf{rel}})$, $K$ number of compensating literals to introduce
**Output:** the relaxed WMI model $(\Delta_+^{\mathsf{rel}}, \mathcal{W}_+^{\mathsf{rel}})$ with compensating literals initialized.

$\Delta_+^{\mathsf{rel}} \leftarrow \Delta^{\mathsf{rel}}, \mathcal{W}_+^{\mathsf{rel}} \leftarrow \mathcal{W}^{\mathsf{rel}}$
$\mathbf{X}_o \leftarrow \mathsf{nonCopyVars}(\mathcal{L})$ ▷ Gather original variables
**for each** $X_i \in \mathbf{X}_o$ **do**
    **for** $k = 1, \ldots, K$ **do**
        $\tau_{i,k} \sim \mathsf{Uniform}(\mathsf{support}(X_i))$ ▷ Randomly help support
        $\sigma_{i,k} \sim \mathsf{Uniform}(\{+1, -1\})$ ▷ And pick one half
        $\ell_{i,k} \leftarrow (X_i \leq \sigma_{i,k} \cdot \tau_{i,k})$
        $\Delta_{+,i}^{\mathsf{rel}} \leftarrow \Delta_{+,i}^{\mathsf{rel}} \wedge \ell_{i,k}$
        $\theta_{i,k} \leftarrow 1$ ▷ Initiate potentials
        $w_{\ell_{i,k}} := \exp(\theta_{i,k})$
        $\mathcal{W}_+^{\mathsf{rel}} \leftarrow \mathcal{W}_+^{\mathsf{rel}} \cup \{w_{\ell_{i,k}}\}$
        **for each** $\ell : (X_i = X_i^c) \in \mathcal{L}$ **do**
            $\ell_{i,k}^c \leftarrow (X_i^c \leq \sigma_{i,k} \cdot \tau_{i,k})$
            $\Delta_{+,i}^{\mathsf{rel},c} \leftarrow \Delta_{+,i}^{\mathsf{rel},c} \wedge \ell_{i,k}$
            $\theta_{i,k}^c \leftarrow 1$ ▷ Initiate potentials
            $w_{\ell_{i,k}^c} := \exp(\theta_{i,k}^c)$
            $\mathcal{W}_+^{\mathsf{rel}} \leftarrow \mathcal{W}_+^{\mathsf{rel}} \cup \{w_{\ell_{i,k}^c}\}$
**Return** $(\Delta_+^{\mathsf{rel}}, \mathcal{W}_+^{\mathsf{rel}})$