CONCEPTTENSOR: A PROBABILISTIC TENSOR REPRESENTATION FOR MULTI-PARENT DAGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Cognitive representation requires encoding entities and their hierarchical relations into a queryable and reasoning-ready formal structure. Existing approaches follow two paths. Symbolic hierarchies are explicit and traceable, but are constrained to a single parent, and cannot express an entity belonging to multiple upper concepts. Distributed vector representations are flexible and generalizable, but are not white-box, and their internal representations are not traceable. White-box, generality, and scalability are difficult to achieve simultaneously: Bayesian networks are the only model that is both white-box and general, yet their exact inference grows exponentially with graph size, and the inference process gradually becomes non-hand-verifiable. We propose ConceptTensor, an architecture that embeds a multi-parent DAG into a high-dimensional tensor space and computes relations directly by tensor contraction. Entity coordinates are determined by a Dirichlet posterior. Hierarchy, parent, and child are carried by three tensor axes. Parent reading, child reading, common-parent score, common-child score, sibling similarity, and multi-hop traversal are all derived directly by tensor contraction. Representation and inference are decoupled; there is no training, and every intermediate value is hand-verifiable. On six query tasks over multi-parent DAGs, the architecture achieves an order-of-magnitude efficiency improvement over Bayesian networks.

1 INTRODUCTION

The core task of cognitive representation is to encode entities and their hierarchical relations in a concept system into a computable, queryable, and reasoning-ready formal structure. Existing methods follow two paths.

Symbolic hierarchies. Tree structures represent hierarchy by parent-child pointers. Their advantage is explicit structure and clear query paths; their disadvantage is that the single-parent constraint is too strong to express an entity belonging to multiple upper concepts. DAGs relax the single-parent constraint, but existing work still treats the DAG directly as the query structure: queries traverse parent-child edges level by level, path length grows with depth, and the semantics of a relation query is determined by the graph traversal algorithm.

Distributed vector representations. High-dimensional embeddings measure concept association by semantic similarity. Their advantage is flexibility and generalizability; their disadvantage is the lack of explicit hierarchy, the inability to directly answer structured queries such as "what is the parent of a concept" or "what is the common ancestor of two concepts," and the fact that internal representations are not traceable.

White-box, generality, and scalability are difficult to achieve simultaneously. Bayesian networks are the only model that is both white-box and general (Pearl, 1988). Their exact inference uses variable elimination (Dechter, 1999), with complexity $O(N \cdot 2^W)$ that grows exponentially with graph size, and the inference process gradually becomes non-hand-verifiable.

We propose ConceptTensor, an architecture that embeds a multi-parent DAG into a high-dimensional tensor space and represents hierarchical and topological relations as probabilistic relations. The core property of the architecture is that both representation and inference are *fully white-box*: entity

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

coordinates are determined directly by level evidence, all relation readings are derived directly by tensor contraction, and every step is hand-verifiable.

1.1 Outsourced Training

The architecture does not involve any training. Inputs are provided by an upstream module: an entity set $\mathcal{E}$, level evidence counts for each entity, and a parent-child relation set $R \subseteq \mathcal{E} \times \mathcal{E}$. Parent-child relations satisfy the level-consistency constraint. The architecture writes these inputs directly into the tensor space and does not learn parameters.

1.2 Contributions

1. We propose ConceptTensor: level probabilities determine entity coordinates; hierarchy and parent-child relations are carried by a hierarchy axis and a relation axis, with the relation axis split into a parent axis and a child axis; all relations are computed directly by tensor contraction.

2. Representation and inference are decoupled: the representation layer carries coordinates and level probabilities, while the inference layer performs tensor contractions independently.

3. Fully white-box: every intermediate value is hand-verifiable from the input, with no training and no hidden parameters.

4. A complete mathematical architecture covering level inference, tensor construction, relation reading, and multi-hop traversal.

5. Traversal depth is determined by the target level; the architecture does not impose a constant upper bound.

6. Comparison with Bayesian networks on multi-parent DAGs across six query tasks, yielding consistent efficiency gains.

2 Related Work

Hierarchy representation. Tree structures represent hierarchy by parent-child pointers with clear paths, but each node has exactly one parent. DAGs relax the single-parent constraint, allowing an entity to have multiple parents. Existing work treats the DAG directly as the query structure: queries traverse parent-child edges level by level, path length grows with depth, and relation queries are bound to the graph structure, with query semantics determined by graph traversal algorithms.

White-box probabilistic models. Bayesian networks represent parent-child relations by a DAG, store probabilities in conditional probability tables (Pearl, 1988), and perform exact inference by variable elimination (Dechter, 1999). DUCG represents parent-child relations by connected event variables and probabilities by causal strengths (Zhang, 2012). Probabilistic concept networks combine concept hierarchy with probabilistic networks (Poh and Fehling, 1993). These models couple the parent-child structure and probability distribution in the same data structure (e.g., CPT); inference steps are determined by this structure, and inference gradually becomes non-hand-verifiable as graph size grows.

Our approach. We embed a multi-parent DAG into a high-dimensional tensor space: hierarchy and parent-child relations live on a hierarchy axis and a relation axis, with the relation axis split into a parent axis and a child axis; parent, child, common parent, common child, sibling, and multi-hop readings are all obtained by tensor contraction on the same set of coordinates. Compared with methods that use the DAG directly as the query structure, our queries are linear-algebraic operations on a tensor, independent of graph traversal algorithms. Compared with white-box probabilistic models, we carry the parent-child structure and the probability distribution separately, and the number of inference steps is determined by the target level.

3 LOGICAL ARCHITECTURE

This section describes the design intent and layer division of ConceptTensor. Formalization follows in Section 4.

3.1 REPRESENTATION LAYER

The representation layer carries two things:

- **Entity coordinates**: each entity has a unique coordinate vector on the hierarchy axis, determined by level evidence via a Dirichlet posterior. Coordinates are fixed once written and do not change across query tasks.
- **Relation tensor**: parent-child relations are written into positions of the relation axis. The level distribution of each edge is spread along the hierarchy axis, producing a three-way non-zero entry.

The representation layer is *static*: once upstream data is written, the representation layer is not updated, and queries do not change it.

3.2 INFERENCE LAYER

The inference layer consists of tensor contractions. A contraction is the act of reading a relation from the tensor: two points' positions in the tensor are maintained by the relation axis, and contraction aligns them on the relation axis to read out their relation. Contraction is not scope-limited and not count-limited. The inference layer has *no state*: it does not write to the representation layer, and does not depend on the previous query's result.

3.3 THREE AXES

From the m axes of the tensor, three are selected to carry relation-reading functions:

- **Hierarchy axis**: carries the level assignment of an entity. Each entity has one probability value at each level.
- **Parent axis**: carries "who points to me."
- **Child axis**: carries "whom I point to."

The parent axis and child axis are jointly referred to as the relation axis. Parent and child are not two independent axes; they are two components of the relation axis, written as two index positions but semantically a single pair of relations.

3.4 DESIGN RATIONALE

Why a tensor. The multi-way indexing of a tensor naturally carries multiple relations. Hierarchy and parent-child live on different axes and do not interfere; contractions along different directions on the same tensor yield different relation readings. There is no need to design a separate data structure for each relation.

Why decouple representation and inference. The representation layer handles "what to store"; the inference layer handles "how to query." They are independent and can be optimized separately: the representation layer can adopt a sparse format, the inference layer can reorder contractions, and neither affects the other.

Why fully white-box. Every non-zero tensor entry is determined directly by level probability, span weight, and edge structure, with no hidden layers, no gradients, and no invisible parameters. Every query reads a position from the tensor and is hand-verifiable.

4 MATHEMATICAL ARCHITECTURE

This section fixes notation, defines objects, and gives propositions.

4.1 PRELIMINARIES

Definition 1 (Level system)**.** Let $L \in \mathbb{N}^+$. The level system is a finite set $\mathcal{L} = \{1, \dots, L\}$, where 1 is the most abstract level and L is the most concrete.

Definition 2 (Entity set and evidence)**.** Let $\mathcal{E} = \{e_1, \dots, e_D\}$ be an entity set with $D = |\mathcal{E}|$. For each $e \in \mathcal{E}$, let $x_1, \dots, x_{N_e}$ be level annotations provided by the upstream module; each x_i assigns e to a unique level $l(x_i) \in \mathcal{L}$. The evidence count is defined as

$$C(e, l) = \sum_{i=1}^{N_e} \mathbf{1}[l(x_i) = l].$$

Note. By Definition 2, $\sum_{l \in \mathcal{L}} C(e, l) = N_e$.

Definition 3 (Level posterior)**.** Take a Dirichlet prior $\theta \sim \mathrm{Dir}(\alpha_1, \dots, \alpha_L)$ with $\alpha_l > 0$, and likelihood $\mathrm{Mult}(N_e, \theta)$ (Bishop, 2006). The posterior mean is

$$P(l \mid e) = \frac{C(e, l) + \alpha_l}{N_e + \sum_{k=1}^{L} \alpha_k}.$$

In the experiments we take $\alpha_l = 1$ (symmetric prior), so the posterior reduces to $(C(e, l)+1)/(N_e+L)$. This normalization ensures that $P(l \mid e)$ is a probability distribution, makes the level distributions of different entities directly comparable, and prevents evidence count from affecting weighted relation quantities.

Lemma 1. For any $e \in \mathcal{E}$, $\sum_l P(l \mid e) = 1$ and $P(l \mid e) > 0$.

Proof. $\sum_l (C(e, l)+\alpha_l) = N_e + \sum_l \alpha_l$, equal to the denominator, so the sum is 1. Positivity follows from $C(e, l) \geq 0$ and $\alpha_l > 0$. □

Definition 4 (Dominant level)**.** For $e \in \mathcal{E}$, the dominant level is

$$\ell^*(e) = \min \left\{ \arg\max_{l \in \mathcal{L}} P(l \mid e) \right\}.$$

The minimum index ensures uniqueness.

Definition 5 (Parent-child relations)**.** The parent-child relation set $R \subseteq \mathcal{E} \times \mathcal{E}$ satisfies

$$(p, c) \in R \implies \ell^*(p) < \ell^*(c).$$

R is a set; the same pair (p, c) appears at most once. If the upstream provides multi-edges, they must be merged first.

Proposition 1 (Acyclicity)**.** $G = (\mathcal{E}, R)$ is a DAG.

Proof. Suppose a cycle $e_{i_1} \to e_{i_2} \to \cdots \to e_{i_k} \to e_{i_1}$ exists. By Definition 5, along every edge the dominant level strictly increases, so $\ell^*(e_{i_1}) < \cdots < \ell^*(e_{i_1})$, a contradiction. □

4.2 TENSOR CONSTRUCTION

Definition 6 (Tensor space and coordinates)**.** The tensor space is

$$\mathcal{H} = \bigotimes_{k=1}^{m} \mathbb{R}^{d_k}, \quad m \geq 3.$$

The coordinate vector $\phi(e)$ of entity $e \in \mathcal{E}$ on the hierarchy axis is defined as the level posterior vector

$$\phi(e) = \big(P(1 \mid e), \dots, P(L \mid e)\big) \in \mathbb{R}^L.$$

The coordinate is fixed once written and does not change across query tasks. The relation tensor $\mathcal{T}$ is the extension of $\phi(e)$ along the relation axis; queries are performed by contractions on $\mathcal{T}$, and $\phi(e)$ itself does not directly participate in contractions.

Definition 7 (Classification axes and tensor shape)**.** The tensor space $\mathcal{H}$ is of order m, $m \geq 3$. From the m axes, three are selected to carry relation-reading functions:

- hierarchy axis a_ℓ of dimension L;
- parent axis a_p of dimension D;
- child axis a_c of dimension D.

The parent and child axes are jointly referred to as the relation axis a_r. The relation tensor $\mathcal{T} \in \mathbb{R}^{L\times D\times D}$ is the order-three slice of $\mathcal{H}$ on these three axes; the remaining $m-3$ axes do not participate in relation reading. All subsequent contractions are performed on $\mathcal{T}$.

Definition 8 (Relation-axis components)**.** The index i of the parent axis a_p corresponds to parent entity $\mathrm{idx}^{-1}(i)$; the index j of the child axis a_c corresponds to child entity $\mathrm{idx}^{-1}(j)$. Parent-child relations are jointly carried by (i, j). In $\mathcal{T}_{l,i,j}$, i takes the parent axis and j takes the child axis.

Definition 9 (Span weight)**.** For $(p, c) \in R$, let $\Delta = \ell^*(c) - \ell^*(p) > 0$, and define

$$w_{p,c}(l) = \exp\left(-\frac{(l-\ell^*(p))^2}{\Delta^2/\ln 2}\right).$$

This weight equals 1 at $l = \ell^*(p)$, equals $1/2$ at $|l - \ell^*(p)| = \Delta$, and decays as a Gaussian with distance.

Definition 10 (Tensor entry)**.** For $(p, c) \in R$, let $i = \mathrm{idx}(p)$, $j = \mathrm{idx}(c)$:

$$\mathcal{T}_{l,i,j} = P(l \mid p) \cdot w_{p,c}(l).$$

For positions $(i, j) \notin R$, we set $\mathcal{T}_{l,i,j} = 0$.

Proposition 2 (Support)**.** $\mathcal{T}_{l,i,j} \neq 0$ if and only if $(e_i, e_j) \in R$.

Proof. Sufficiency: if $(e_i, e_j) \in R$, by Lemma 1, $P(l \mid e_i) > 0$; by Definition 9, $w_{e_i,e_j}(l) > 0$. Hence the product is positive. Necessity: if $(e_i, e_j) \notin R$, by Definition 10 the entry is 0. Therefore $\mathcal{T}_{l,i,j} \neq 0$ if and only if $(e_i, e_j) \in R$. □

Proposition 3 (Non-negativity and Frobenius bound)**.** $\mathcal{T}_{l,i,j} \geq 0$. Let $\Delta_{out} = \max_i |\{j : (e_i, e_j) \in R\}|$. Then

$$\|\mathcal{T}\|_F^2 \leq L \cdot D \cdot \Delta_{out}.$$

Proof. Non-negativity is immediate from $P, w \geq 0$. For the bound: $w \leq 1$ by Definition 9; $P \leq 1$ by Lemma 1, so $\mathcal{T}_{l,i,j} \leq 1$. For fixed i, the number of non-zero (i, j) is $\leq \Delta_{out}$; for fixed (i, j), the number of non-zero entries is L. Hence the total number of non-zero entries is $\leq L \cdot D \cdot \Delta_{out}$. Each squared entry is ≤ 1, and summing gives $\|\mathcal{T}\|_F^2 \leq L \cdot D \cdot \Delta_{out}$. □

4.3 Contraction and Relation Reading

Definition 11 (Contraction)**.** Let $\mathcal{T} \in \mathbb{R}^{L\times D\times D}$. The sum contraction along the hierarchy axis is

$$\mathrm{Sum}_\ell(\mathcal{T})_{i,j} = \sum_{l=1}^{L} \mathcal{T}_{l,i,j}.$$

The inner-product contraction along the hierarchy axis is

$$\mathrm{Inn}_\ell(\mathcal{T}; \mathcal{T}')_{i,j} = \sum_{l=1}^{L} \mathcal{T}_{l,i,j} \cdot \mathcal{T}'_{l,i,j}.$$

The partial inner product along the hierarchy axis is

$$\mathrm{PartInn}_\ell(\mathcal{T}; \mathcal{T}')_{i,j,j'} = \sum_{l=1}^{L} \mathcal{T}_{l,i,j} \cdot \mathcal{T}'_{l,i,j'}.$$

When $\mathcal{T} = \mathcal{T}'$, we abbreviate to $\mathrm{PartInn}_\ell(\mathcal{T})_{i,j,j'}$. All subsequent relation reading is given by combinations of these three contractions. Contraction is not scope-limited and not count-limited.

Definition 12 (Parent reading)**.** For $e_c \in \mathcal{E}$, let $j = \mathrm{idx}(c)$:

$$\mathrm{Parents}(e_c) = \left\{ e_p \;\middle|\; \sum_{l=1}^{L} \mathcal{T}_{l,\mathrm{idx}(p),j} > 0 \right\}.$$

Definition 13 (Child reading)**.** For $e_p \in \mathcal{E}$, let $i = \mathrm{idx}(p)$:

$$\mathrm{Children}(e_p) = \left\{ e_c \;\middle|\; \sum_{l=1}^{L} \mathcal{T}_{l,i,\mathrm{idx}(c)} > 0 \right\}.$$

Proposition 4 (Reading duality)**.** For any $(p, c) \in R$, $c \in \mathrm{Children}(p)$ if and only if $p \in \mathrm{Parents}(c)$.

Proof. By Proposition 1, $\mathcal{T}_{l,\mathrm{idx}(p),\mathrm{idx}(c)} \neq 0$ if and only if $(p, c) \in R$. This condition is symmetric for parent reading and child reading on the support-set criterion. □

4.4 Probabilistic Relations

Definition 14 (Common-parent score)**.** For $e_{c_1}, e_{c_2} \in \mathcal{E}$, let $j_1 = \mathrm{idx}(c_1)$, $j_2 = \mathrm{idx}(c_2)$:

$$\mathrm{CP}(c_1, c_2)[p] = \mathrm{PartInn}_\ell(\mathcal{T})_{\mathrm{idx}(p),j_1,j_2}.$$

This quantity is a level-weighted score, not normalized, used to determine common parents and to carry level weights.

Proposition 5. $\mathrm{CP}(c_1, c_2)[p] > 0$ if and only if $p \in \mathrm{Parents}(c_1) \cap \mathrm{Parents}(c_2)$.

Proof. By Proposition 1, $\mathcal{T}_{l,\mathrm{idx}(p),j_1} > 0$ for some l if and only if $(p, c_1) \in R$, and $\mathcal{T}_{l,\mathrm{idx}(p),j_2} > 0$ if and only if $(p, c_2) \in R$. The product-sum is positive if and only if both hold. □

Definition 15 (Common-child score)**.** For $e_{p_1}, e_{p_2} \in \mathcal{E}$, let $i_1 = \mathrm{idx}(p_1)$, $i_2 = \mathrm{idx}(p_2)$:

$$\mathrm{CC}(p_1, p_2)[c] = \mathrm{PartInn}_\ell(\mathcal{T})_{i_1,i_2,\mathrm{idx}(c)}.$$

This quantity is a level-weighted score, not normalized.

Proposition 6. $\mathrm{CC}(p_1, p_2)[c] > 0$ if and only if $c \in \mathrm{Children}(p_1) \cap \mathrm{Children}(p_2)$.

Proof. Symmetric to Proposition 3. □

Definition 16 (Parent weight vector)**.** For $e \in \mathcal{E}$, let $j = \mathrm{idx}(e)$:

$$w_e[k] = \sum_{l=1}^{L} \mathcal{T}_{l,k,j}, \quad k = 1, \ldots, D,$$

where k ranges over indices of all candidate parent entities.

Definition 17 (Sibling similarity)**.** For $e_i, e_j \in \mathcal{E}$:

$$S_{i,j} = \frac{\sum_k \min(w_i[k], w_j[k])}{\sum_k \max(w_i[k], w_j[k])}.$$

If the denominator is 0, we set $S_{i,j} = 0$. This is a level-probability-weighted Jaccard similarity (Jaccard, 1901) taking values in $[0, 1]$.

Proposition 7. $S_{i,j} \in [0, 1]$ and $S_{i,j} = S_{j,i}$.

Proof. The numerator and denominator are symmetric in i, j, giving symmetry. For boundedness: $\min \leq \max$ term-by-term, so the denominator is at least as large as the numerator after summing. Hence $S_{i,j} \in [0, 1]$. □

4.5 MULTI-HOP TRAVERSAL

Definition 18 (Ancestor set)**.**

$$\mathrm{Anc}^{(0)}(e) = \{e\}, \quad \mathrm{Anc}^{(k+1)}(e) = \bigcup_{e' \in \mathrm{Anc}^{(k)}(e)} \mathrm{Parents}(e').$$

By Proposition 7, when $k > \ell^*(e) - 1$, $\mathrm{Anc}^{(k)}(e)$ stops growing.

Definition 19 (Descendant set)**.**

$$\mathrm{Desc}^{(0)}(e) = \{e\}, \quad \mathrm{Desc}^{(k+1)}(e) = \bigcup_{e' \in \mathrm{Desc}^{(k)}(e)} \mathrm{Children}(e').$$

By symmetry, when $k > L - \ell^*(e)$, $\mathrm{Desc}^{(k)}(e)$ stops growing.

Proposition 8 (Traversal depth)**.** If $e_t \in \mathrm{Anc}^{(k)}(e_s)$ and $e_t \neq e_s$, then

$$k \leq \ell^*(e_s) - \ell^*(e_t).$$

Proof. By Definition 5, each parent-child step strictly decreases the dominant level. By the recursive definition of $\mathrm{Anc}^{(k)}$, there is a path of length k from e_s to e_t on which every step strictly decreases the level. Hence $k \leq \ell^*(e_s) - \ell^*(e_t)$. □

4.6 SINGLE-TARGET QUERY

Definition 20 (Single-target query)**.** For $e_s, e_t \in \mathcal{E}$, query whether e_t is a parent of e_s:

$$\mathrm{Query}(e_s, e_t) = \sum_{l=1}^{L} \mathcal{T}_{l,\mathrm{idx}(t),\mathrm{idx}(s)}.$$

This query returns the level-weighted score of e_t as a parent of e_s.

Proposition 9. $\mathrm{Query}(e_s, e_t) > 0$ if and only if $e_t \in \mathrm{Parents}(e_s)$.

Proof. By Proposition 1, $\mathcal{T}_{l,\mathrm{idx}(t),\mathrm{idx}(s)} \neq 0$ if and only if $(t, s) \in R$. The sum is positive if and only if $(t, s) \in R$. □

4.7 REACHABILITY

Definition 21 (Reachability)**.** Let $e_s, e_t \in \mathcal{E}$. If there exists $k \geq 0$ such that $e_t \in \mathrm{Anc}^{(k)}(e_s) \cup \mathrm{Desc}^{(k)}(e_s)$, we say e_s and e_t are reachable. The global search query returns this Boolean determination.

5 EXPERIMENTS

5.1 SETUP

Datasets. Two multi-parent DAGs:

Dataset	N	L	Level distribution
DAG-25	25	5	1/3/5/7/9
DAG-50	50	5	1/5/10/15/19

Each child randomly selects 1–2 parents with strictly smaller level. Random seed is fixed at 42. Each entity carries a 5-dimensional level evidence count; the dominant level is inferred by the Dirichlet posterior.

Baselines. Bayesian network, built with `pgmpy` (Ankan and Panda, 2015), with exact inference by variable elimination (Dechter, 1999). BN edges are filtered by the same level-consistency constraint

as ConceptTensor, so that support sets match. Structural queries on the BN side use CSR sparse matrices (Saad, 2003) to avoid pgmpy wrapper overhead contaminating the speedup.

Query tasks.

Task	Description	Reference
T1 Global search	Reachability between random S and T	Def. 21
T2 Sibling similarity	Weighted Jaccard	Def. 17
T3 Common parent	Common-parent score	Def. 14
T4 Common child	Common-child score	Def. 15
T5 Ancestor set	All ancestors	Def. 18
T6 Descendant set	All descendants	Def. 19

Metric. We report results for $N = 25$, 100 repetitions per task. Speedup is defined as BN time divided by CT time.

5.2 Results

Task	CT (ms)	BN (ms)	Speedup
T1 Global search	94.53	572.31	6.05×
T2 Sibling similarity	9.43	58.49	6.20×
T3 Common parent	5.26	56.44	10.72×
T4 Common child	5.12	55.44	10.84×
T5 Ancestor set	17.01	130.55	7.67×
T6 Descendant set	17.21	131.87	7.66×

5.3 Analysis

1. **Double-tensor contraction tasks (T3, T4) achieve the highest speedup**, at 10.72× and 10.84×. CT uses the partial inner product along the hierarchy axis, and BN uses CSR intersections.
2. **Single-tensor contraction tasks (T2, T5, T6) achieve 6.20× to 7.67×**. CT performs a sum along the hierarchy axis, while BN performs CSR indexing or BFS.
3. **Global search (T1) achieves 6.05×**.
4. **BN-side CSR queries return Boolean support sets, while CT returns level-weighted scores; CT still achieves the above speedups while additionally carrying level weights.**

5.4 Complexity Comparison

Task	ConceptTensor	Bayesian Network
Parent query	$O(L \cdot \Delta_{in})$	$O(\Delta_{in})$
Common parent	$O(L \cdot D)$	$O(\Delta_{in})$
Multi-hop ancestor	$O(L^2 \cdot D)$ dense	$O(L \cdot E)$ sparse

The above is asymptotic complexity; Table 1 is measured time. CT's advantage comes from vectorized contraction along the hierarchy axis and from constant factors; at the small scale of $L = 5$, dense representation is not a bottleneck. CT's advantage lies in carrying level probability weights and in deriving weighted scores directly by double-tensor contraction, rather than Boolean sets, not in raw structural query speed.

6 Discussion

White-box property. Every non-zero value of $\mathcal{T}_{l,i,j}$ is determined by $P(l \mid p)$, $w_{p,c}(l)$, and the edge (p, c), with no hidden layers, no gradients, and no invisible parameters. Query is contraction; failure

is diagnosable. This is the essential difference between ConceptTensor and embedding models or GNNs.

Relation to same-tier frameworks. ConceptTensor, Bayesian networks (Pearl, 1988), DUCG (Zhang, 2012), and probabilistic concept networks (Poh and Fehling, 1993) all belong to the family of pure white-box probabilistic cognitive representation frameworks. Differences: parent-child relations are carried by the tensor relation axis, probabilities by the level probability distribution, relation reading is given directly by tensor contraction, and multi-hop inference depth is determined by the target level.

Role of normalization. The only normalization in this paper appears in Definition 3: $P(l \mid e)$ is a normalized level probability distribution. CP, CC, and $S_{i,j}$ are all weighted scores or similarities based on this probability; they are not themselves normalized probability distributions.

7 Conclusion

We proposed ConceptTensor, a probabilistic tensor representation architecture for multi-parent DAGs. Entity coordinates are determined by a Dirichlet posterior and do not change across queries; parent-child relations are carried by a hierarchy axis and a relation axis, with the relation axis split into a parent axis and a child axis; all relation readings are derived directly by tensor contraction. The architecture fully decouples representation and inference, with no training. On six query tasks over multi-parent DAGs, the architecture achieves an order-of-magnitude efficiency improvement over Bayesian networks.

AI use statement

In this work, we used DeepSeek for the following tasks: (i) assisting with mathematical derivations, including the formulation of the tensor construction rules and the proofs of the propositions in Section 4; (ii) implementing the experimental code and the query operators used in Section 5; (iii) polishing and editing the manuscript to improve readability and language; and (iv) searching for relevant literature and background materials. We have reviewed all AI-assisted work: the mathematical derivations and proofs were checked against the definitions in Section 4; the AI-assisted code was reviewed, tested, and debugged by the authors, and its outputs were verified against the mathematical definitions; the polished text was verified for factual accuracy; the literature found through AI-assisted search was manually reviewed and only cited when relevant. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Reproducibility statement

All definitions, propositions, and proofs are given in Section 4. The experimental code, data generation scripts, and query implementations will be made available as supplementary material. The datasets used in the experiments are generated with a fixed random seed (42), and the generation procedure is fully described in Section 5.1.

References

Ankan, A. and Panda, A. pgmpy: Probabilistic graphical models using Python. In *Proceedings of the 14th Python in Science Conference (SciPy 2015)*, pp. 6–11, 2015.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Dechter, R. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113(1–2):41–85, 1999.

Jaccard, P. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

Poh, K.-L. and Fehling, M. R. Probabilistic conceptual network: A belief representation scheme for utility-based categorization. In *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI 1993)*, pp. 211–218, 1993.

Saad, Y. *Iterative Methods for Sparse Linear Systems*, 2nd edition. SIAM, 2003.

Zhang, Q. Dynamic uncertain causality graph for knowledge representation and reasoning: Discrete DAG cases. *Journal of Computer Science and Technology*, 27(1):1–23, 2012.

A SYMBOL TABLE

Symbol	Meaning
$\mathcal{E}$	Entity set
$\mathcal{L}$	Level system
$\mathcal{H}$	High-dimensional tensor space
$\phi(e)$	Coordinate vector of e on the hierarchy axis, i.e., the level posterior $P(l \mid e)$
$P(l \mid e)$	Level posterior
$\ell^*(e)$	Dominant level
R	Parent-child relation set
$\mathcal{T}$	Relation tensor
a_ℓ, a_p, a_c	Hierarchy, parent, and child axes
Δ_{out}	Maximum out-degree of a node
Sum_ℓ	Sum contraction along the hierarchy axis
Inn_ℓ	Inner-product contraction along the hierarchy axis
$\mathrm{PartInn}_\ell$	Partial inner product along the hierarchy axis
$\mathrm{Parents}(e)$	Parent set
$\mathrm{Children}(e)$	Child set
$\mathrm{CP}(c_1, c_2)$	Common-parent score
$\mathrm{CC}(p_1, p_2)$	Common-child score
$S_{i,j}$	Sibling similarity
$\mathrm{Anc}^{(k)}(e)$	k-hop ancestor set
$\mathrm{Desc}^{(k)}(e)$	k-hop descendant set

B TERMINOLOGY

- **High-dimensional tensor space**: an order-m tensor with $m \geq 3$. Entity coordinates are written onto the hierarchy axis; the relation tensor is an order-three slice.
- **Three axes**: hierarchy axis, parent axis, child axis. The parent and child axes are jointly referred to as the relation axis.
- **Contraction**: the act of reading a relation from the tensor. Not scope-limited and not count-limited.
- **Probabilistic relations**: common-parent score, common-child score, sibling similarity. All are weighted by the level posterior $P(l \mid e)$.
- **Normalization**: appears only in Definition 3, and operates on the level distribution of a single entity, not on similarities between entities.