CERTIFIED CONSTRUCTIVE BOX EMBEDDINGS FOR DAG TAXONOMIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Embeddings of a taxonomy are usually trained. However, containment holds only up to a residual error, and the geometry reflects the frequencies in the training data. A geometry compiled from the order structure alone can guarantee two properties that no trained embedding offers: certified zero violations on a declared pair family, and retrieval rankings that no corpus frequency has shaped. We build on GRAIL, a classic reachability index that labels each node of a DAG with one interval per random linear extension, and we lift this labelling into the representation space. Instead of drawing extensions at random, we select each embedding dimension as a *witness* that separates specific pairs from a declared disjointness family. The resulting box embedding has *certified zero* containment and separation violations on that family, attested by a machine-checkable certificate that any third party can re-verify. Randomised GRAIL labels remain nonzero even at four times the dimension budget, and gradient-trained box baselines leave residual violations even when trained to fit the full graph. Because the geometry never sees a corpus, ranking by box volume never buries rare, specific concepts on GO, MeSH, or WordNet, while every similarity-based ranker prefers popular ones, at retrieval utility matching an oracle prefilter. Construction takes seconds, and full Gene Ontology under P_2, with 3.6×10^8 separation pairs, builds in $28\,\mathrm{s}$. The containment slack between two boxes also predicts held-out edges. Compared with fairly tuned baselines on three datasets, its AUC is higher on the densely multi-parent MeSH, on par on GO, and lower on the nearly tree-shaped WordNet. Symbolic graph features remain above every embedding method. With no task-specific training, the same slack tracks graded lexical entailment on HyperLex at $\rho{=}0.761$.

1 INTRODUCTION

Hypernymy detection, taxonomy completion, and hierarchy-aware retrieval rely on geometric representations of human-curated DAG taxonomies such as WordNet, the Gene Ontology, and MeSH. Axis-aligned boxes (Vilnis et al., 2018; Dasgupta et al., 2020) are a natural primitive: containment encodes parent–child entailment, and volume gives concept specificity. All of these box embeddings are trained by gradient descent on a violation loss. However, gradient training cannot guarantee perfect containment and converges to a different solution on every run. A downstream system therefore either tolerates an unknown violation rate or re-validates every query against the symbolic graph.

Consider a retrieval prefilter that must return every descendant of a query concept. An unknown 0.1% error rate turns it into a heuristic. The construction provides the one-sided, checkable guarantee it needs: no true *is-a* is ever rejected, and no pair from the declared family is falsely accepted, so the filter composes with downstream stages without re-validation. The guarantee is also auditable: a third party can re-run the certificate check mechanically, whereas a trained embedding has no declared policy to certify against.

For plain reachability queries on a static graph, a classical index is exact and remains the right tool: GRAIL with post-filter traversal (Yıldırım et al., 2010), a closure table, or a breadth-first search. Our construction does not compete with these indexes. It addresses questions a graph lookup cannot answer. Many systems need the taxonomy in the same vector space as their text embeddings, so that hierarchy and similarity are scored in one batched operation. They ask graded questions: how specific

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.

a concept is, how strongly one term entails another (Section 7), and which concepts to retrieve without frequency bias (Section 6). They must handle edges the visible graph does not contain (Section 7). And they often cannot ship a graph runtime, only arrays and a certificate that any consumer can re-check. A certified geometry keeps the exactness of the index on the declared family and adds these capabilities.

Our approach is to compile the geometry instead of training it. GRAIL (Yıldırım et al., 2010) labels each node of a DAG with d intervals, one per random linear extension (Dushnik & Miller, 1941), and ancestors' intervals contain descendants' by construction. We keep this labelling and replace the random draw. Each dimension is selected as a *witness extension* that separates specific pairs from a *declared* disjointness family (Theorem 2), and compatible pairs are packed greedily into shared dimensions (Theorem 3). This addresses the extension-selection problem that GRAIL leaves open, for a structured relaxation of it: instead of chasing every false positive, the practitioner declares a family from the chain $P_0 \subset P_1 \subseteq P_{1.5}(r) \subseteq P_2 \subseteq P_3$ (Section 3.1), and the construction drives violations on that family to exactly zero, certified by a machine-checkable artifact (Section 3.4). The construction runs in seconds on ontologies of 10^4–10^5 concepts (Section 4).

The combinatorial machinery in this work is classical, and the interval labelling follows GRAIL (Section 2). Our contributions are what certified selection adds on top. (i) Witness selection over a declared disjointness family, and the resulting zero-violation guarantee that random selection does not reach (Sections 3, 5). (ii) The policy chain with its radius parameter r, where $P_{1.5}(r{=}2)$ saturates structural MAP at a fraction of the dimensional cost of P_2 (Section 4). (iii) A certification primitive that makes the guarantee machine-checkable, and a violation audit of trained box baselines under the same families (Section 3.4). (iv) A retrieval-bias analysis. Compiled from the order structure alone, the geometry cannot inherit a corpus's frequency bias: volume ranking has popularity correlation $\rho \leq 0$ on every dataset, every similarity-based ranker we test has $\rho > 0$, and retrieval utility matches an oracle prefilter (Section 6). (v) A generalisation analysis. On held-out edges the slack leads fairly tuned learned baselines in AUC on MeSH, ties them on GO, and falls behind on the almost tree-shaped WordNet, following how much multi-parent structure each hierarchy has, with symbolic graph features as the ceiling (Section 7). The same slack tracks graded lexical entailment on HyperLex (Vulić et al., 2017) without task-specific training (Section 7).[1]

2 Related Work

Geometric embeddings of hierarchies come in three main families. Order embeddings (Vendrov et al., 2016) encode entailment as coordinate-wise dominance, hyperbolic embeddings (Nickel & Kiela, 2017; Ganea et al., 2018) exploit the exponential volume growth of hyperbolic space for tree-like structure, and axis-aligned box embeddings (Vilnis et al., 2018; Dasgupta et al., 2020) encode parent–child entailment as containment and concept specificity as volume. We adopt the box primitive. Later work extends boxes to joint hierarchies, entity typing, knowledge-graph reasoning, and capacity analyses (Patel et al., 2020; Onoe et al., 2021; Chen et al., 2021; Abboud et al., 2020; Boratko et al., 2021). Closest to our guarantee axis are box embeddings of $\mathcal{EL}^{++}$ ontologies, BoxEL (Xiong et al., 2022) and Box^{2}EL (Jackermeier et al., 2024), where faithfulness holds at the loss optimum and no per-artifact certificate is emitted. Except for the tree-restricted n-ball construction of Dong et al. (2019), all of these methods train on losses that penalise violations without eliminating them.

A 2023–2026 wave of taxonomy-editing work uses boxes as a learned target, from BoxTaxo (Jiang et al., 2023) for taxonomy expansion through completion, topic-taxonomy discovery, ontology extension, and uncertainty-aware expansion (Xue et al., 2024; Zhang et al., 2024; Lu et al., 2024; Memariani et al., 2025; Mishra et al., 2026), with BoxTaxo still a standard baseline in LLM-era pipelines (Zeng et al., 2025; Mishra et al., 2025). These models learn the taxonomy, and expansion conditions on the text of a concept that has no edges yet. Ours takes the taxonomy as input and realises it with a certificate, so Section 7 compares against structure-only baselines trained on the same visible graph rather than against these text-conditioned systems.

[1] Code, build scripts, and documentation at `https://github.com/anonymous-institute/constructive_boxes.git` (anonymised for review).

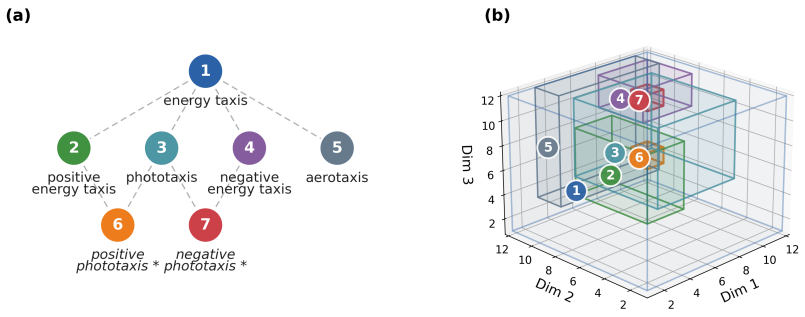

Figure 1: Constructive box embedding of a small GO subgraph (*energy taxis*, n=7). **(a)** the input DAG; **(b)** the resulting axis-aligned boxes in three dimensions. Each parent–child edge is realised as exact box containment in every dimension simultaneously. Siblings under a common parent are disjoint in at least one dimension by Theorem 2. Pairs marked $*$ illustrate P_2-only co-hyponym separations inherited from a non-root common ancestor.

The construction we build on comes from the reachability-labelling literature. GRAIL (Yıldırım et al., 2010) assigns each node one interval per random linear extension, giving the one-sided guarantee our Theorem 1 restates, and resolves the remaining false positives by falling back to graph traversal. Successors such as FELINE (Veloso et al., 2014) select their orders deliberately. Our delta is threefold: the labels are consumed as a geometric representation instead of a filter in front of a traversal engine, the extension-selection problem is solved for a *declared* pair family instead of all false positives (where the NP-hardness of false-positive minimisation bites), and the result carries a certificate instead of a fallback path. We make no speed claims against these systems.

In spatial indexing, nested axis-aligned regions date to the R-tree (Guttman, 1984) and its successors (Beckmann et al., 1990; Berchtold et al., 1996; Ciaccia et al., 1997), which reduce sibling overlap heuristically. Our construction enforces sibling disjointness under a declared policy instead.

On the combinatorial side, Dushnik & Miller (1941) define the order dimension of a poset as the minimum number of linear extensions whose intersection recovers it. Deciding whether it exceeds two is NP-complete (Yannakakis, 1982). The dimension count our packer minimises is the *separation dimension* of Basavaraju et al. (2016), restricted to linear extensions of the poset and applied to the hypergraph of descendant blocks $\{\mathrm{block}(v) : v \in V\}$. Classical set-cover bounds (Johnson, 1974; Chvátal, 1979; Feige, 1998) apply unchanged. The contribution of this paper is the applied construction, namely the policy chain, the packing heuristic over witness extensions, and the certification artefact. We do not compute the minimum-dimensional realiser.

3 Constructive Box Embeddings

Let $(V, \leq)$ be a finite partial order with root, where nodes are concepts and $u \leq v$ means u is a descendant of v (or $u = v$). Write $\mathrm{desc}(v)$ for the strict descendants of v and $\mathrm{block}(v) = \mathrm{desc}(v) \cup \{v\}$. A *box embedding* is a map $B : V \to \mathbb{R}^d$ assigning each node an axis-aligned box $B(v) = \prod_{i=1}^{d} [\ell_i(v), h_i(v)]$.

We are interested in two structural properties. *Containment* (P): for each parent–child edge $v \to u$, $B(u) \subseteq B(v)$, meaning $\ell_i(v) \leq \ell_i(u)$ and $h_i(u) \leq h_i(v)$ in every dimension i. *Separation* (DC, under family S): for each pair $(a, b) \in S$, $B(a) \cap B(b) = \emptyset$, for which interval disjointness in one dimension suffices. Containment is mandatory for every edge. Separation is policy-dependent: which incomparable pairs to enforce is the construction's main design choice (Section 3.1).

The per-dimension labelling we use is GRAIL's (Yıldırım et al., 2010): one linear extension per dimension, with node v's interval running from its rank to the maximum rank over $\mathrm{block}(v)$, so that containment holds in every dimension by construction. Everything after the labelling differs from GRAIL: we select each extension as a witness against the declared family (Section 3.2) and consume the output as a certified geometric representation (Section 2).

3.1 SEPARATION POLICIES

Choosing S trades DC coverage against dimensionality: a stronger policy separates more incomparable pairs but generally requires more dimensions. We call a pair (a, b) *constructively separable* when $\mathrm{block}(a) \cap \mathrm{block}(b) = \emptyset$, that is, neither node is an ancestor of the other and their descendant sets are disjoint. If the two blocks overlap, any interval assigned to one must also cover the shared node and so cannot be disjoint from the other's interval. Only constructively-separable pairs admit a witness extension under this construction (Theorem 2). We organise the candidate families into a chain $P_0 \subset P_1 \subseteq P_{1.5}(r) \subseteq P_2 \subseteq P_3$:

- P_0 (NONE): $S = \emptyset$, containment only.
- P_1 (SIBLING): siblings under a common parent with disjoint descendants.
- $P_{1.5}(r)$ (LOCAL COMMON ANCESTOR): P_1 plus every constructively-separable pair (a, b) for which some non-root node x reaches a and b through two different children, each along a directed path of at most r edges (shortest directed distance, so a node with several paths from x qualifies through its shortest one). $P_{1.5}(1) = P_1$ and $P_{1.5}(\infty) = P_2$, so r smoothly interpolates between the two.
- P_2 (COMMON ANCESTOR): P_1 plus every constructively-separable pair that shares a non-root common ancestor.
- P_3 (ALL INCOMPARABLE): every constructively-separable pair.

P_3 is not enumerated here: with 5×10^8 to 3×10^9 candidate pairs it is feasible on GO and MeSH at a multiple of the P_2 cost and exceeds memory on WordNet, where P_2 already does (Appendix A). P_2 is the strongest policy we construct, and it drives structural false positives near zero relative to P_1 (Section 4). We use P_2 where memory permits and the tunable $P_{1.5}(r)$ otherwise. On WordNet the latter is the only currently usable approximation beyond P_1 (Limitations).

3.2 CONSTRUCTION ALGORITHM

Each dimension of the embedding corresponds to one *linear extension* L of $(V, \leq)$, a total order respecting $u \leq v \Rightarrow u$ later than v. Given L, node v's interval in that dimension is

$$\ell(v) = \mathrm{rank}_L(v), \quad h(v) = \max_{u \in \mathrm{block}(v)} \mathrm{rank}_L(u). \tag{1}$$

Ancestors automatically contain descendants: an ancestor starts no later in the order, and its block extends at least as far.

Extensions are constructed via priority-biased Kahn's algorithm: a topological sort in which the next-visited node is the available one with the smallest priority. Choosing priorities -1 on $\mathrm{block}(a)$, $+1$ on $\mathrm{block}(b)$, and 0 elsewhere produces an extension that places every node of $\mathrm{block}(a)$ before any node of $\mathrm{block}(b)$, giving disjoint intervals (Theorem 2).

A naive realisation uses one dimension per pair in S, giving $d = |S|$. Many pairs are co-separable, so a single extension witnesses several at once, and we exploit this with a greedy multi-target packer: at each step, choose a candidate extension that covers the largest set of still-uncovered pairs, commit it as one dimension, iterate. This is a set-cover heuristic over S, and on real ontologies the packing factor is 30 to 10^6 pairs per dimension (Section 4). The full procedure is Algorithm 1.

The dominant cost is evaluating candidate extensions. Each candidate costs one priority-biased topological sort, $O(|V| + |E|)$, plus one interval test for every pair still in U. Two implementation details omitted from Algorithm 1, a merge step that adds further uncovered pairs to a seed's priority assignment and a direct-targeting phase for the last pairs, are described in Appendix B. Neither affects the corollary in Section 3.3. The inner loop is compiled in Cython and parallelised via OpenMP, so candidate evaluation scales near-linearly with core count, and end-to-end wall-clock is reported in Section 4.

The construction is complete at this point, but intervals coming straight from rank positions can leave aspect ratios in the thousands and per-edge slack near zero. An optional four-phase post-construction optimiser therefore refines the geometry within the feasible region (Appendix C). We report both *Constructive* and *Constructive + optimiser* in Section 7, with the latter re-certified under the same policy.

Algorithm 1: Construction of a $0\,P + 0\,DC$ box embedding

Input : DAG $(V, \leq)$; separation policy P
Output : boxes $B : V \to \mathbb{R}^d$ with P=0, DC=0 on S

$S \leftarrow$ SEPARATIONPAIRS$(V, \leq, P)$; // enumerate pairs the policy enforces
$\mathcal{L} \leftarrow [\,]$; $U \leftarrow S$; // committed extensions; uncovered pairs
while $U \neq \emptyset$ **do**
  sample $S' \subseteq U$ uniformly without replacement, $|S'| = \min(20,\ \max(5,\ \lfloor |U|/500 \rfloor))$;
  foreach $(a, b) \in S'$ **do**
    $L_{(a,b)} \leftarrow$ KAHN$(V, \leq$; priorities -1 on block(a), $+1$ on block$(b))$;
  $L \leftarrow \arg\max_{L' \in \{L_{(a,b)} : (a,b) \in S'\}} |\{(a', b') \in U : L' \text{ separates block}(a'), \text{block}(b')\}|$;
  $C \leftarrow \{(a', b') \in U : L \text{ separates block}(a'), \text{block}(b')\}$;
  $\mathcal{L}$.append(L); $U \leftarrow U \setminus C$;
for *dimension* i, *node* v **do**
  $\ell_i(v) \leftarrow \text{rank}_{L_i}(v)$; $h_i(v) \leftarrow \max_{u \in \text{block}(v)} \text{rank}_{L_i}(u)$;
return $B(v) = \prod_{i=1}^{|\mathcal{L}|} [\ell_i(v),\ h_i(v)]$;

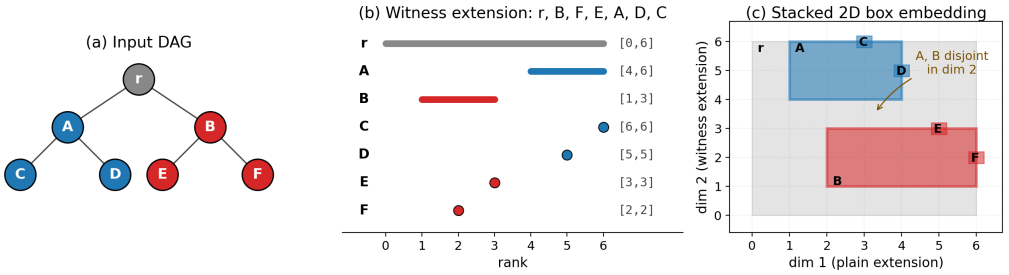

Figure 2: The construction in two dimensions. **(a)** A six-node DAG. **(b)** Priority-biased Kahn's with priority -1 on block(B) visits all of B's block first and produces a witness extension with disjoint intervals B:$[1, 3]$ and A:$[4, 6]$, where a plain extension such as r, A, B, C, D, E, F leaves the siblings overlapping (A:$[1, 4]$, B:$[2, 6]$). **(c)** Stacking a plain and a witness extension yields a box embedding with every ancestor pair contained in every dimension and (A, B) disjoint in dimension 2.

3.3 CORRECTNESS

All results in this subsection hold for any DAG and any separation family of constructively-separable pairs.

Theorem 1 (Containment correctness). *For any linear extension L of $(V, \leq)$, the interval assignment of Equation* (1) *satisfies: if $u \leq v$ then $[\ell(u), h(u)] \subseteq [\ell(v), h(v)]$.*

Containment costs nothing: an ancestor starts no later than its descendant in any extension, and its block reaches at least as far, so both interval endpoints nest. Separation is what needs a deliberately chosen extension.

Theorem 2 (Witness extension). *For any constructively-separable pair (a, b), there exists a linear extension L of $(V, \leq)$ in which $[\ell(a), h(a)]$ and $[\ell(b), h(b)]$ are disjoint.*

The witness exists because block-disjointness keeps the DAG acyclic after adding every edge from block(a) to block(b). Any topological order of the augmented DAG places the two blocks one after the other, which is exactly what the priority-biased Kahn's algorithm of Section 3.2 produces. Since containment holds under every extension, stacking witness extensions cannot break it, and the two properties compose.

Theorem 3 (Realiser composition). *Let $S = \{(a_1, b_1), \ldots, (a_k, b_k)\}$ be a finite family of constructively-separable pairs and let $L_1, \ldots, L_k$ be witness extensions for each pair (Theorem 2). The k-dimensional box embedding obtained by giving each node its $[\ell, h]$ interval under*

each L_i simultaneously realises (i) containment for every $u \leq v$ in every dimension, and (ii) interval-disjointness for every pair $(a_i, b_i) \in S$ in dimension i.

Corollary 1. *Algorithm 1 maintains the invariant that a pair is removed from U only after some committed extension separates it. Therefore, configured with separation family S, it produces a box embedding with $P = 0$ and $DC = 0$ on S for any DAG. The greedy packer affects only the dimension count, not correctness, and compresses the $|S|$ witness extensions into $d \leq |S|$ dimensions.*

3.4 CERTIFICATION

The guarantee is mechanically verifiable. The certification procedure (a) re-enumerates the separation family S under the declared policy, (b) re-loads the saved embedding B, and (c) checks $B(u) \subseteq B(v)$ for every edge $u \to v$ and $B(a) \cap B(b) = \emptyset$ in at least one dimension for every $(a, b) \in S$. Step (c) is a parallelised per-pair scan with early exit. The output is a signed JSON artefact recording the embedding hash, the declared policy, the pair count $|S|$, and P=0, DC=0. It asserts zero violations *on the $|S|$ pairs of the declared policy* and nothing outside S. Systems that need a stronger guarantee re-certify under a stronger policy.

4 SCALING AND POLICY COVERAGE

Table 1 sweeps the policy chain $P_0 \subset P_1 \subset P_{1.5}(r) \subset P_2$ on GO, MeSH, and WordNet at full scale. The multi-target packer makes the sweep affordable: it compresses the 357 million P_2 pairs on GO into 292 dimensions. P_1 already cuts mean false positives per query from five digits into single digits at 44 to 101 dimensions. Where P_2 is feasible it removes another two to three orders of magnitude (GO $4.94 \to 0.011$) at roughly three times the dimensions. GO P_2 enumerates $3.6{\times}10^8$ pairs and is the only configuration whose build needs more than a gigabyte of memory.

Within the $P_{1.5}(r)$ family, structural MAP saturates at r=2, within 0.001 of P_2 on GO and MeSH. Larger r keeps driving mean FP towards the P_2 floor at the cost of more dimensions, so r acts as a deployment-time knob between certified coverage and dimension budget. We use $P_{1.5}(r{=}2)$ for every constructive embedding in the paper.

The table separates the certified construction from the optional refinement optimiser (Appendix C). The construction stays under 8 s everywhere except GO under P_2, which compiles its 3.6×10^8-pair family in 28 s. Refinement improves aspect ratios and edge slack, costs up to 162 s on that largest cell, and is never required for the guarantee. Certification and every zero-violation claim in this paper attach to the construction alone.

5 CERTIFIED ZERO VERSUS ALMOST ZERO

Certified zero on a declared family is the property that separates the construction from its relatives. Table 2 quantifies the gap on full GO from two directions: trained box models evaluated by our validator, and randomised GRAIL labellings at growing dimension budgets.

The top block runs our validator over the geometries of the two closest trained relatives, trained on the *full* graph so that memorisation is allowed. Both leave five-to-seven-digit residuals on every family we check. The learned box model keeps 41,778 containment violations. The Gumbel box model keeps 305,675 sibling-disjointness violations and 3.2 million on $P_{1.5}(r{=}2)$. These are the residuals of a loss optimum, so further tuning is not guaranteed to remove them, and nothing in training says where they sit. They also carry a cost: ranking descendants through the trained geometries collapses structural MAP to 0.378 and 0.003, against 0.912 for every interval labelling in the table. The construction reaches all three zeros in four seconds, against 58 and 160 seconds of training for the two baselines, and the certificate states the checked family explicitly.

GRAIL draws its linear extensions at random, so every added dimension reduces false positives on average but targets none in particular. The middle block shows that even at d=768, four times our budget, randomised labels retain 97 violations of the sibling family and 288 of $P_{1.5}(r{=}2)$, where the witness construction reaches certified zero at d=191. The MAP column is flat across all labellings, so the benefit of targeted selection lies entirely in the guarantee.

6 FREQUENCY-NEUTRAL RETRIEVAL

Similarity-based retrieval inherits the frequency statistics of its training corpus: popular concepts sit in dense neighbourhoods and get retrieved first, and rare, specific concepts sink. A geometry compiled from the taxonomy alone has no corpus to inherit from, and we test whether that translates into measurably frequency-neutral retrieval.

For each dataset we issue every concept at depth at least three with at least five descendants as a query and rank its retrieved descendants. The text embedding of a concept is the mean of the 50-d GloVe vectors (Pennington et al., 2014) of the tokens in its name (first lemma for WordNet synsets). We compare six rankers: cosine similarity over these vectors, similarity behind an oracle prefilter that removes all non-descendants, the box-containment candidate set in canonical node order, a hybrid that ranks the box-containment candidates by similarity, a graph oracle that returns the true descendant set in canonical order, and volume ranking, which orders the box-containment candidates by box volume so that specific concepts come first. Every ranker returns at most 100 candidates and MAP is normalised by the full descendant set, so even the oracles fall below 1 on queries with more than 100 descendants, and the three box-based rankers share a candidate set and hence a MAP. Popularity is measured by proxies independent of the graph structure: PubMed occurrence counts for MeSH, annotation counts for GO, and corpus frequency for WordNet. ρ_{indep} is the Spearman correlation between a concept's rank position and its popularity under these proxies, where positive values mean popular concepts are retrieved first.

Table 3 shows one sign pattern without exception. Volume ranking is the only ranker with $\rho_{\text{indep}} \leq 0$ on every dataset (GO -0.006, WordNet -0.173, MeSH -0.275), and every similarity-based ranker has $\rho_{\text{indep}} > 0$ on every dataset (up to $+0.200$ on MeSH). Volume ranking never buries the tail, and similarity always does. The neutrality costs no utility: volume ranking's MAP is within 0.001 of the oracle prefilter on GO and WordNet and sits 0.008 below it on MeSH. On GO the descendant-count proxy ρ_{desc} is partly structural (propagated annotation counts include descendants'), so the claim rests on ρ_{indep}.

7 HELD-OUT EDGE PREDICTION AND GRADED ENTAILMENT

We now test whether the compiled geometry generalises to edges it has never seen. We hold out 10% of non-tree edges from each of GO, MeSH, and WordNet, build the constructive embedding on the remaining 90%, and rank held-out positives against two negative-sampling modes: *random* non-descendants (the conventional easy baseline) and *sibling* (a node sharing a direct parent with

Dataset	Policy	$\lvert S\rvert$	d	MAP	Mean FP	Constr. (s)	Refine (s)
GO	P_0	0	1	.001	12,462	0.4	0.4
	P_1	697,578	93	.904	4.939	1.1	4.1
$\lvert V\rvert$=38,560	$P_{1.5}(r{=}2)$	**5,572,393**	**191**	**.912**	**0.037**	**4.1**	**12.2**
	P_2	356,707,671	292	.913	0.011	28.1	162.1
MeSH	P_0	0	1	.001	12,653	0.3	0.3
	P_1	255,662	101	.943	1.078	0.9	4.2
$\lvert V\rvert$=31,108	$P_{1.5}(r{=}2)$	**2,564,874**	**142**	**.952**	**0.023**	**2.1**	**6.2**
	P_2	29,379,446	175	.953	0.004	3.4	24.4
WordNet	P_0	0	1	.001	27,320	0.9	0.6
	P_1	1,839,265	44	.912	15.038	1.5	4.5
$\lvert V\rvert$=82,115	$P_{1.5}(r{=}2)$	**12,685,210**	**73**	**.914**	**0.065**	**7.8**	**9.2**
	$P_{1.5}(r{=}4)$	138,535,652	106	.915	0.002	61.7	83.0

Table 1: Separation policies on WordNet (Miller, 1995), Gene Ontology (The Gene Ontology Consortium, 2000), and MeSH (Lipscomb, 2000). The bold $r{=}2$ rows are the operating point used throughout the paper, where structural MAP saturates. Omitted radii interpolate smoothly, and WordNet P_2 and $r{\geq}6$ exceed memory (Limitations). Mean FP is the average non-descendant count per containment query. Times from a 20-core workstation split the certified construction from the optional refinement (Appendix C), and the certificate holds for the raw construction.

Method	d	MAP	P	DC (P_1)	DC ($P_{1.5}(r{=}2)$)
Learned box (Vilnis et al., 2018)	191	.378	41,778	3,075	726,777
Gumbel box (Dasgupta et al., 2020)	191	.003	1,940	305,675	3,195,567
GRAIL (random) (Yıldırım et al., 2010)	93	.912	$\mathbf{0}^{\checkmark}$	1,364	5,945
	191	.913	$\mathbf{0}^{\checkmark}$	506	1,932
	384	.913	$\mathbf{0}^{\checkmark}$	227	819
	768	.913	$\mathbf{0}^{\checkmark}$	97	288
Constructive (P_1 cert.)	93	.904	$\mathbf{0}^{\checkmark}$	$\mathbf{0}^{\checkmark}$	6,452
Constructive ($P_{1.5}(r{=}2)$ cert.)	191	.912	$\mathbf{0}^{\checkmark}$	$\mathbf{0}^{\checkmark}$	$\mathbf{0}^{\checkmark}$

Table 2: Violations of the declared separation families on full GO (697,578 P_1 pairs and 5,572,393 $P_{1.5}(r{=}2)$ pairs). The trained baselines fit the *entire* graph under the tuned regime of Section 7, so memorisation is allowed, and their violations collapse structural descendant retrieval (MAP column). Every interval labelling has $P{=}0$ structurally (Theorem 1), yet randomised GRAIL labels stay nonzero on the DC families even at four times the constructive budget (the pair-seeded variant behaves similarly), while witness selection reaches zero on all three families at $d{=}191$ at the same MAP, attested by the machine-checkable certificate of Section 3.4. The P_1-certified row shows that the certificate covers exactly the declared family.

	MAP			ρ_{indep}			$\rho^{\dagger}_{\text{desc}}$		
Ranker	GO	MeSH	WN	GO	MeSH	WN	GO	MeSH	WN
Similarity	.189	.109	.071	.046	.031	.038	.105	.026	.063
Oracle prefilter + sim	.957	.999	.952	.065	.200	.123	.282	.146	.172
Box containment (canonical order)	.956	.991	.952	.185	.038	.011	.193	.053	.019
Hybrid	.956	.995	.952	.065	.197	.123	.281	.145	.172
Volume ranking (ours)	.956	.991	.952	$\mathbf{-.006}$	$\mathbf{-.275}$	$\mathbf{-.173}$	$-.783$	$-.688$	$-.689$
Graph oracle	.957	.999	.952	.185	.039	.011	.194	.054	.019

Table 3: Frequency neutrality of hierarchy-aware retrieval (WN: WordNet). ρ_{indep}: Spearman correlation between rank quality and a frequency-independent popularity proxy, where positive means popular concepts are retrieved first. $\rho^{\dagger}_{\text{desc}}$: correlation with descendant count, partly tautological for the volume ranker since volume is monotone in descendant count, so ρ_{indep} is the load-bearing column. Volume ranking is the only method with $\rho_{\text{indep}} \leq 0$ on every dataset, at MAP within 0.008 of the oracle prefilter (italics: oracles with access to the true descendant set).

the true child, the structural class the policy enforces on the training graph). Spanning-tree edges are protected so the training graph stays connected, and gradient baselines are trained on the same 90% subgraph at matched dimensionality. The constructive scorer is the minimum per-dimension containment slack $\min_d\big(\text{hw}_d^{\text{par}} - |c_d^{\text{par}} - c_d^{\text{ch}}| - \text{hw}_d^{\text{ch}}\big)$, positive when the child fits strictly inside the parent and negative otherwise.

Hypernymy detection on the full input WordNet is theorem-covered for the policy classes (ten-seed table in Appendix E).

The comparison in Table 4 is set up in the baselines' favour: they train minibatched with early stopping on the transitive closure of the visible graph, Poincaré uses its paper's burn-in and is-a scoring, dimensionality is matched to the construction's, and all ranking metrics use average-rank tie handling.

(1) *The compiled slack wins where multi-parent structure is dense, and loses where the hierarchy is nearly a tree.* Against the strongest learned baseline, order embeddings, the constructive slack leads on MeSH by $+0.035$ (random) and $+0.030$ (sibling) AUC ($p \leq 2\times10^{-8}$, paired over ten seeds), is 0.005 behind on GO in both modes, and is clearly overtaken on WordNet (-0.21 random, -0.12 sibling). The ordering tracks structure: MeSH and GO are heavily multi-parent (30% and 39% of nodes), where the witness construction spends its dimensions, while WordNet's noun hierarchy is 97% tree, where a soft order embedding compresses the structure more efficiently. The post-construction

Method	GO		MeSH		WordNet	
	rand	sib	rand	sib	rand	sib
Constructive (ours)	$.911_{\pm.002}$	$.788_{\pm.002}$	$.804_{\pm.005}$	$.726_{\pm.004}$	$.638_{\pm.000}$	$.603_{\pm.000}$
+ optimiser	$\mathbf{.918}_{\pm.003}$	$\mathbf{.796}_{\pm.002}$	$\mathbf{.821}_{\pm.004}$	$\mathbf{.738}_{\pm.004}$	$.625_{\pm.001}$	$.596_{\pm.000}$
Order emb. (Vendrov et al., 2016)	$.916_{\pm.004}$	$.793_{\pm.003}$	$.769_{\pm.007}$	$.696_{\pm.005}$	$\mathbf{.843}_{\pm.012}$	$\mathbf{.719}_{\pm.006}$
Poincaré (Nickel & Kiela, 2017)	$.783_{\pm.005}$	$.689_{\pm.004}$	$.671_{\pm.005}$	$.614_{\pm.004}$	$.556_{\pm.011}$	$.571_{\pm.006}$
Gumbel box (Dasgupta et al., 2020)	$.591_{\pm.003}$	$.548_{\pm.003}$	$.603_{\pm.020}$	$.572_{\pm.014}$	$.771_{\pm.003}$	$.663_{\pm.003}$
Learned box (Vilnis et al., 2018)	$.595_{\pm.002}$	$.582_{\pm.002}$	$.465_{\pm.004}$	$.487_{\pm.004}$	$.492_{\pm.004}$	$.493_{\pm.003}$
Cones (Ganea et al., 2018)	$.722_{\pm.035}$	$.642_{\pm.021}$	$.749_{\pm.018}$	$.672_{\pm.013}$	$.655_{\pm.029}$	$.594_{\pm.015}$
BoxE (Abboud et al., 2020)	$.760_{\pm.004}$	$.675_{\pm.003}$	$.660_{\pm.006}$	$.614_{\pm.005}$	$.764_{\pm.009}$	$.668_{\pm.007}$

Table 4: Held-out edge prediction AUC, ten seeds per cell ($x_{\pm.yyy}$ means $x \pm 0.\text{yyy}$). **rand**: random non-descendant negatives. **sib**: visible-sibling negatives, topped up from the random pool when a positive has fewer than 50 siblings. The compiled slack leads on multi-parent MeSH, ties on GO, and trails on almost tree-shaped WordNet. Best value per column in bold. The symbolic ceiling (Jaccard-anc. AUC $.974$ / $.862$ / $.905$ under random negatives) and full metrics are in Appendix D.

optimiser adds $+0.007$ to $+0.017$ AUC on GO and MeSH and is neutral on WordNet. Rank-aware metrics are less favourable: order embeddings lead AP and MRR under random negatives on GO and MeSH, and the gap closes only under sibling negatives (Appendix D).

(2) *The slack scorer is doing the work, not naive containment.* A binary contains-vs-not scorer over the same embedding scores at chance (0.50 AUC, 0.04 MRR, Appendix D). This is expected: the embedding compiles the visible 90% exactly and nothing more, so a child whose edge was withheld is never inside its parent's box. What carries the signal is the size of the violation. The withheld child still shares the parent's other ancestors, so in most dimensions its interval sits close to the parent's and sticks out only slightly. An unrelated node sticks out far. The signed slack measures exactly this distance and therefore ranks the true child above the unrelated node, although both are outside.

(3) *Symbolic graph features remain the upper bound.* Jaccard overlap of ancestor sets sits above every geometric method on every dataset and mode (random-mode AUC 0.974 / 0.862 / 0.905, full rows in Appendix D). The gap to that ceiling is 0.041 to 0.062 AUC, the price of compressing the graph into a fixed-dimensional geometry that buys volume, slack, and composability in return. The symbolic features inspect the visible graph directly, so we treat them as a ceiling, not a competing embedding.

The same slack also carries graded entailment. HyperLex (Vulić et al., 2017) grades $2{,}616$ word pairs by entailment strength. We evaluate on its $2{,}163$ noun pairs, map lemmas to candidate synsets via NLTK (Bird et al., 2009), score each synset pair by the signed L^∞ slack between their boxes, and aggregate per pair with MAX. Neither the construction nor the optimiser sees a HyperLex label. The score correlates with the human grade at $\rho{=}0.741$ Spearman for the raw construction and $\rho{=}0.761$ after the optimiser, so the compiled geometry carries graded entailment information without any task-specific training. For context, Nickel & Kiela (2017) report $\rho{=}0.512$ for Poincaré embeddings of WordNet on the same $2{,}163$ noun pairs, 0.389 for a Euclidean embedding, and 0.21 to 0.28 for the WordNet-path and frequency baselines of Vulić et al. (2017), under their own lemma-to-synset protocol.

8 Conclusion

A direct combinatorial construction compiles any DAG into axis-aligned box embeddings with certified zero violations on a declared policy, in seconds even at 10^5 concepts. Training does not provide this guarantee: trained box baselines and randomised GRAIL labels both keep residual violations. Because the geometry sees only the order structure, volume ranking retrieves without frequency bias at oracle-level utility. On held-out edges the compiled slack leads fairly tuned structure-only baselines in AUC where multi-parent structure is dense, trails the best of them on top-rank precision under random negatives, and tracks graded entailment without task-specific training.

ETHICS STATEMENT

Application scope. The construction operates on existing human-curated taxonomies (WordNet, GO, MeSH). It does not generate or augment those taxonomies. Biases or omissions in the upstream resources propagate through to the embedding unchanged. Downstream systems that rely on the embedding for high-stakes decisions (clinical coding, legal retrieval) should audit the underlying taxonomy independently.

Compute and environmental cost. The full experimental suite (build and certify all separation policies on three ontologies, ten-seed hypernymy on WordNet, ten-seed held-out edge prediction on GO/MeSH/WordNet, HyperLex, policy and radius ablations, incremental benchmark) runs in $\sim$9 hours on a single 20-core workstation with one RTX 3090. The GPU serves the gradient baselines. The largest construction (GO P_2) is CPU-only and one-shot and takes under four minutes including refinement (Table 1). Ten-seed held-out edge prediction dominates the rest of the budget at $\sim$5.3 hours.

Use of large language models. In line with the ICLR policy on large language model usage, we disclose that coding and writing assistance was provided by Claude Fable 5 (Anthropic). Earlier iterations of the codebase and manuscript were assisted by GPT 5.5 (OpenAI) and Claude Opus 4.7 (Anthropic). LLM assistance covered drafting, refactoring, and editing. All technical claims, theorems, proofs, experimental designs, and conclusions are the authors' own, no LLM is an author, and the authors take full responsibility for the entire content of the paper.

REPRODUCIBILITY STATEMENT

All results are reproducible from the anonymised repository referenced in Section 1: a single driver script regenerates every results file behind the tables of this paper from the raw ontology files, each results file carries a provenance sidecar recording the git commit, configuration, and seeds (ten seeds for every stochastic experiment), and every zero-violation claim ships as a machine-checkable JSON certificate that the repository's verification tool re-checks independently of the construction code (Section 3.4).

REFERENCES

Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. BoxE: A box embedding model for knowledge base completion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6dbbe6abe5f14af882ff977fc3f35501-Abstract.html.

Manu Basavaraju, L. Sunil Chandran, Martin Charles Golumbic, Rogers Mathew, and Deepak Rajendraprasad. Separation dimension of graphs and hypergraphs. *Algorithmica*, 75(1):187–204, 2016. doi: 10.1007/s00453-015-0050-6.

Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The R*-Tree: An efficient and robust access method for points and rectangles. In *ACM SIGMOD International Conference on Management of Data*, 1990. doi: 10.1145/93597.98741.

Stefan Berchtold, Daniel A. Keim, and Hans-Peter Kriegel. The X-Tree: An index structure for high-dimensional data. In *Proceedings of VLDB*, 1996. URL https://www.vldb.org/conf/1996/P028.PDF.

Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

Michael Boratko, Dongxu Zhang, Nicholas Monath, Luke Vilnis, Shikhar Murty, Rajarshi Das, and Andrew McCallum. Capacity and bias of learned geometric embeddings for directed graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/88d25099b103efd638163ecb40a55589-Abstract.html.

Xuelu Chen, Michael Boratko, Shib Sankar Dasgupta, Xiang Lorraine Li, and Andrew McCallum. Probabilistic box embeddings for uncertain knowledge graph reasoning. In *Proceedings of NAACL-HLT*, 2021. URL https://aclanthology.org/2021.naacl-main.68/.

Vasek Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979. doi: 10.1287/moor.4.3.233.

Paolo Ciaccia, Marco Patella, and Pavel Zezula. M-Tree: An efficient access method for similarity search in metric spaces. In *Proceedings of VLDB*, 1997. URL https://www.vldb.org/conf/1997/P426.PDF.

Shib Sankar Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Lorraine Li, and Andrew McCallum. Improving local identifiability in probabilistic box embeddings. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/01c9d2c5b3ff5cbba349ec39a570b5e3-Abstract.html.

Tiansi Dong, Christian Bauckhage, Hailong Jin, Juanzi Li, Olaf Cremers, Daniel Speicher, Armin B. Cremers, and Joerg Zimmermann. Imposing category trees onto word-embeddings using a geometric construction. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=rJlWOj0qF7.

Ben Dushnik and E. W. Miller. Partially ordered sets. *American Journal of Mathematics*, 63(3): 600–610, 1941. doi: 10.2307/2371374.

Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998. doi: 10.1145/285055.285059.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1646–1655. PMLR, 2018. URL https://proceedings.mlr.press/v80/ganea18a.html.

Antonin Guttman. R-Trees: A dynamic index structure for spatial searching. In *ACM SIGMOD International Conference on Management of Data*, 1984. doi: 10.1145/971697.602266.

Mathias Jackermeier, Jiaoyan Chen, and Ian Horrocks. Dual box embeddings for the description logic EL++. In *Proceedings of the ACM Web Conference (WWW)*, pp. 2250–2258, 2024.

Song Jiang, Qiyue Yao, Qifan Wang, and Yizhou Sun. A single vector is not enough: Taxonomy expansion via box embeddings. In *Proceedings of the ACM Web Conference (WWW)*, 2023. doi: 10.1145/3543507.3583310.

David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, 1974. doi: 10.1016/S0022-0000(74)80044-9.

Carolyn E. Lipscomb. Medical subject headings (MeSH). *Bulletin of the Medical Library Association*, 88(3):265–266, 2000.

Yuyin Lu, Hegang Chen, Pengbo Mao, Yanghui Rao, Haoran Xie, Fu Lee Wang, and Qing Li. Self-supervised topic taxonomy discovery in the box embedding space. *Transactions of the Association for Computational Linguistics (TACL)*, 2024. doi: 10.1162/tacl_a_00712.

Adel Memariani, Martin Glauer, Simon Flügel, Fabian Neuhaus, Janna Hastings, and Till Mossakowski. Box embeddings for extending ontologies: A data-driven and interpretable approach. *Journal of Cheminformatics*, 2025. doi: 10.1186/s13321-025-01086-1.

George A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11): 39–41, 1995. doi: 10.1145/219717.219748.

Sahil Mishra, Kumar Arjun, and Tanmoy Chakraborty. Rank, chunk and expand: Lineage-oriented reasoning for taxonomy expansion. In *Findings of the Association for Computational Linguistics: ACL*, 2025. URL https://aclanthology.org/2025.findings-acl.671/.

Sahil Mishra, Srinitish Srinivasan, Srikanta Bedathur, and Tanmoy Chakraborty. TaxoBell: Gaussian box embeddings for self-supervised taxonomy expansion. In *Proceedings of the ACM Web Conference 2026 (WWW)*, pp. 4326–4337, 2026. URL https://dl.acm.org/doi/abs/10.1145/3774904.3792674.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/59dfa2df42d9e3d41f5b02bfc32229dd-Abstract.html.

Yasumasa Onoe, Michael Boratko, Andrew McCallum, and Greg Durrett. Modeling fine-grained entity types with box embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021. URL https://aclanthology.org/2021.acl-long.160/.

Dhruvesh Patel, Shib Sankar Dasgupta, Michael Boratko, Xiang Lorraine Li, Luke Vilnis, and Andrew McCallum. Representing joint hierarchies with box embeddings. In *Automated Knowledge Base Construction (AKBC)*, 2020. URL https://openreview.net/forum?id=J246NSqR_l.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

The Gene Ontology Consortium. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000. doi: 10.1038/75556.

Renê Rodrigues Veloso, Loïc Cerf, Wagner Meira Jr., and Mohammed J. Zaki. Reachability queries in very large graphs: A fast refined online search approach. In *Proceedings of the 17th International Conference on Extending Database Technology (EDBT)*, pp. 511–522, 2014.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. In *International Conference on Learning Representations (ICLR)*, 2016. URL https://arxiv.org/abs/1511.06361.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. Probabilistic embedding of knowledge graphs with box lattice measures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018. URL https://aclanthology.org/P18-1025/.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. HyperLex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4):781–835, 2017. URL https://aclanthology.org/J17-4004/.

Bo Xiong, Nico Potyka, Trung-Kien Tran, Mojtaba Nayyeri, and Steffen Staab. Faithful embeddings for EL++ knowledge bases. In *Proceedings of the 21st International Semantic Web Conference (ISWC)*, pp. 22–38, 2022.

Wei Xue, Yongliang Shen, Wenqi Ren, Jietian Guo, and Jiawei Han. Insert or attach: Taxonomy completion via box embedding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. URL https://aclanthology.org/2024.acl-long.212/.

Mihalis Yannakakis. The complexity of the partial order dimension problem. *SIAM Journal on Algebraic Discrete Methods*, 3(3):351–358, 1982. doi: 10.1137/0603036.

Hilmi Yıldırım, Vineet Chaoji, and Mohammed J. Zaki. GRAIL: Scalable reachability index for large graphs. *Proceedings of the VLDB Endowment*, 3(1):276–284, 2010.

Qingkai Zeng, Yuyang Bai, Zhaoxuan Tan, Zhenyu Wu, Shangbin Feng, and Meng Jiang. CodeTaxo: Enhancing taxonomy expansion with limited examples via code language prompts. In *Findings of the Association for Computational Linguistics: ACL*, pp. 4131–4144, 2025. URL https://aclanthology.org/2025.findings-acl.214/.

Yuhang Zhang, Jiwei Qin, and Chongren Feng. PEB-TAXO: Projecting entities as boxes for taxonomy expansion. *Neural Processing Letters*, 56(2):102, 2024. doi: 10.1007/s11063-024-11575-3.

A LIMITATIONS

The construction assumes a finite input DAG. It does not discover the hierarchy from text, so applications to corpora without an existing taxonomy need an upstream extraction or alignment step that we do not address. For the same reason the held-out-edge comparison of Section 7 is against structure-only geometric baselines and not against taxonomy-expansion systems that condition on concept text.

The greedy packer gives an upper bound on the separation-realiser dimension, not a tight value. Computing the optimum is at least as hard as set cover (Feige, 1998), and on rooted trees under P_1 a single DFS preorder dimension is optimal, which the generic packer misses. Special-casing DFS on trees is straightforward, and closing the gap on general DAGs is open.

The 1.000 AUC on full-WordNet hypernymy is only partly theorem-covered. Positives and policy-covered cohyponym negatives are guaranteed by Theorems 1 and 2. Random-incomparable negatives outside the policy are not, and their empirically perfect separation is a property of WordNet at $d{=}73$ rather than a corollary (Appendix E).

Our incremental builder preserves the guarantee under edge insertions but provides no speedup yet. On GO and MeSH at P_2 with 10% of edges held out it runs within 2% of full rebuild time, because the packer rebuilds the witness extensions for every pair the inserted edge touches. A truly incremental algorithm with a witness-extension dependency tracker is open future work.

Building WordNet P_2 exhausts the $64\,\text{GB}$ of our workstation, as does $P_{1.5}(r{\geq}6)$. The binding constraint is downstream memory, and a streaming audit of the placer and optimiser, or a sparser pair representation, would lift it. The recommended operating point $P_{1.5}(r{=}2)$ is unaffected, and radii up to 4 remain tractable (Table 1).

B IMPLEMENTATION DETAILS OF ALGORITHM 1

The released implementation follows Algorithm 1 with two additions. First, before a seed's extension is scored, it greedily adds up to $2{,}000$ further uncovered pairs to the same priority assignment, in a random order, and keeps the merged extension if it covers at least as many pairs as the seed alone. Second, once at most 20 pairs remain, the bulk phase ends and each remaining pair is targeted directly with its own witness extension. Both only change which extensions are proposed. A pair still leaves U only when a committed extension separates it, so the corollary in Section 3.3 is unaffected. The seed sample S' is drawn uniformly without replacement with $|S'| = \min(20, \max(5, \lfloor |U|/500 \rfloor))$, and all random draws are seeded from the iteration index, so the construction is reproducible for a given input and policy.

C POST-CONSTRUCTION OPTIMISER

The construction yields a feasible solution with $0\,P + 0\,DC$ but does not optimise box geometry: intervals come straight from rank positions, so aspect ratios can be in the thousands and per-edge slack can be zero. The optional optimiser refines geometry within the feasible region in four phases. (A) bottom-up shrink with a fixed per-edge slack reserve and a minimum width on every node; (B) widen parent boundaries to maximise the minimum slack across edges (SVM-style maximum margin); (C) constraint projection with asymmetric buffers (full slack bottom-up, minimal top-down) to avoid oscillation on multi-parent subgraphs; (D) targeted residual fix that re-derives separating dimensions from the post-A/B/C geometry and projects containment plus separation with a 10^{-6} buffer. Phases A and B are heuristic, leaving intermediate violations on synthetic DAGs. Phases C and D restore $(0\,P, 0\,DC)$ on the standard ontologies. End-state validation is the load-bearing check: we re-certify the optimised embedding under the same policy as the raw construction and reject any embedding with residual violations. The raw construction is saved and certified separately, so a rejected optimiser output leaves a valid, certified embedding in place. Aspect ratios drop from 10^2–10^3 to near 1, and slack-based downstream scorers (HyperLex L^∞ slack and link prediction in Section 7) gain a few points, while the guarantee itself is unchanged.

D ADDITIONAL METRICS FOR HELD-OUT EDGE PREDICTION

Table 5 reports AP, MRR, and Hits under random negatives and Table 6 under sibling negatives, ten seeds per dataset from the same runs as Table 4. Three patterns are worth noting beyond what AUC already shows.

Method	AUC	AP	MRR	Hits@1	Hits@10
GO					
Constructive (ours)	$.911_{\pm.002}$	$.253_{\pm.006}$	$.632_{\pm.006}$	$.518_{\pm.008}$	$.850_{\pm.006}$
+ optimiser	$\mathbf{.918}_{\pm.003}$	$.319_{\pm.007}$	$.654_{\pm.008}$	$.542_{\pm.009}$	$.871_{\pm.008}$
Constructive (binary)	$.500_{\pm.000}$	$.020_{\pm.000}$	$.039_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$
+ optimiser (binary)	$.521_{\pm.001}$	$.056_{\pm.001}$	$.078_{\pm.001}$	$.041_{\pm.001}$	$.041_{\pm.001}$
Order emb. (Vendrov et al., 2016)	$.916_{\pm.004}$	$\mathbf{.452}_{\pm.006}$	$\mathbf{.721}_{\pm.007}$	$\mathbf{.614}_{\pm.007}$	$\mathbf{.911}_{\pm.006}$
Poincaré (Nickel & Kiela, 2017)	$.783_{\pm.005}$	$.042_{\pm.001}$	$.295_{\pm.013}$	$.149_{\pm.019}$	$.585_{\pm.011}$
Gumbel box (Dasgupta et al., 2020)	$.591_{\pm.003}$	$.035_{\pm.001}$	$.251_{\pm.008}$	$.118_{\pm.007}$	$.527_{\pm.007}$
Learned box (Vilnis et al., 2018)	$.595_{\pm.002}$	$.025_{\pm.000}$	$.170_{\pm.002}$	$.079_{\pm.002}$	$.346_{\pm.006}$
Cones (Ganea et al., 2018)	$.722_{\pm.035}$	$.064_{\pm.013}$	$.427_{\pm.039}$	$.253_{\pm.034}$	$.822_{\pm.043}$
BoxE (Abboud et al., 2020)	$.760_{\pm.004}$	$.100_{\pm.004}$	$.488_{\pm.006}$	$.354_{\pm.008}$	$.759_{\pm.008}$
Jaccard-anc.	$.974_{\pm.001}$	$.484_{\pm.007}$	$.763_{\pm.006}$	$.617_{\pm.007}$	$.978_{\pm.002}$
Adamic-Adar	$.960_{\pm.001}$	$.515_{\pm.004}$	$.732_{\pm.004}$	$.567_{\pm.005}$	$.960_{\pm.004}$
Common-anc.	$.950_{\pm.001}$	$.394_{\pm.002}$	$.721_{\pm.005}$	$.547_{\pm.005}$	$.959_{\pm.004}$
Shortest-path	$.500_{\pm.000}$	$.020_{\pm.000}$	$.038_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$
MeSH					
Constructive (ours)	$.804_{\pm.005}$	$.120_{\pm.005}$	$.418_{\pm.007}$	$.295_{\pm.008}$	$.681_{\pm.008}$
+ optimiser	$\mathbf{.821}_{\pm.004}$	$.135_{\pm.005}$	$.445_{\pm.008}$	$.321_{\pm.009}$	$.711_{\pm.007}$
Constructive (binary)	$.500_{\pm.000}$	$.020_{\pm.000}$	$.039_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$
+ optimiser (binary)	$.503_{\pm.001}$	$.024_{\pm.001}$	$.044_{\pm.001}$	$.005_{\pm.001}$	$.005_{\pm.001}$
Order emb. (Vendrov et al., 2016)	$.769_{\pm.007}$	$\mathbf{.225}_{\pm.007}$	$\mathbf{.486}_{\pm.005}$	$\mathbf{.371}_{\pm.005}$	$.695_{\pm.009}$
Poincaré (Nickel & Kiela, 2017)	$.671_{\pm.005}$	$.029_{\pm.001}$	$.192_{\pm.009}$	$.085_{\pm.011}$	$.411_{\pm.016}$
Gumbel box (Dasgupta et al., 2020)	$.603_{\pm.020}$	$.038_{\pm.006}$	$.295_{\pm.015}$	$.161_{\pm.012}$	$.567_{\pm.020}$
Learned box (Vilnis et al., 2018)	$.465_{\pm.004}$	$.018_{\pm.000}$	$.075_{\pm.003}$	$.019_{\pm.002}$	$.137_{\pm.006}$
Cones (Ganea et al., 2018)	$.749_{\pm.018}$	$.094_{\pm.008}$	$.469_{\pm.015}$	$.292_{\pm.016}$	$\mathbf{.862}_{\pm.011}$
BoxE (Abboud et al., 2020)	$.660_{\pm.006}$	$.045_{\pm.003}$	$.292_{\pm.006}$	$.180_{\pm.005}$	$.516_{\pm.010}$
Jaccard-anc.	$.862_{\pm.003}$	$.297_{\pm.009}$	$.549_{\pm.006}$	$.407_{\pm.008}$	$.788_{\pm.006}$
Adamic-Adar	$.808_{\pm.003}$	$.297_{\pm.008}$	$.497_{\pm.007}$	$.345_{\pm.008}$	$.649_{\pm.006}$
Common-anc.	$.806_{\pm.003}$	$.230_{\pm.007}$	$.491_{\pm.007}$	$.331_{\pm.008}$	$.650_{\pm.006}$
Shortest-path	$.500_{\pm.000}$	$.020_{\pm.000}$	$.038_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$
WordNet					
Constructive (ours)	$.638_{\pm.000}$	$.133_{\pm.002}$	$.292_{\pm.002}$	$.203_{\pm.003}$	$.457_{\pm.003}$
+ optimiser	$.625_{\pm.001}$	$.132_{\pm.003}$	$.295_{\pm.002}$	$.210_{\pm.003}$	$.452_{\pm.003}$
Constructive (binary)	$.500_{\pm.000}$	$.020_{\pm.000}$	$.038_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$
+ optimiser (binary)	$.500_{\pm.000}$	$.020_{\pm.000}$	$.038_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$
Order emb. (Vendrov et al., 2016)	$\mathbf{.843}_{\pm.012}$	$\mathbf{.232}_{\pm.008}$	$.476_{\pm.013}$	$\mathbf{.332}_{\pm.008}$	$.785_{\pm.023}$
Poincaré (Nickel & Kiela, 2017)	$.556_{\pm.011}$	$.020_{\pm.001}$	$.125_{\pm.008}$	$.054_{\pm.007}$	$.230_{\pm.013}$
Gumbel box (Dasgupta et al., 2020)	$.771_{\pm.003}$	$.137_{\pm.005}$	$\mathbf{.478}_{\pm.009}$	$.304_{\pm.009}$	$\mathbf{.821}_{\pm.011}$
Learned box (Vilnis et al., 2018)	$.492_{\pm.004}$	$.019_{\pm.000}$	$.082_{\pm.003}$	$.020_{\pm.003}$	$.164_{\pm.007}$
Cones (Ganea et al., 2018)	$.655_{\pm.029}$	$.051_{\pm.011}$	$.313_{\pm.037}$	$.140_{\pm.023}$	$.695_{\pm.059}$
BoxE (Abboud et al., 2020)	$.764_{\pm.009}$	$.091_{\pm.007}$	$.327_{\pm.014}$	$.192_{\pm.011}$	$.618_{\pm.023}$
Jaccard-anc.	$.905_{\pm.000}$	$.299_{\pm.004}$	$.555_{\pm.002}$	$.370_{\pm.004}$	$.878_{\pm.001}$
Adamic-Adar	$.896_{\pm.000}$	$.276_{\pm.003}$	$.490_{\pm.002}$	$.283_{\pm.004}$	$.843_{\pm.001}$
Common-anc.	$.884_{\pm.000}$	$.196_{\pm.002}$	$.490_{\pm.002}$	$.283_{\pm.004}$	$.843_{\pm.001}$
Shortest-path	$.500_{\pm.000}$	$.020_{\pm.000}$	$.038_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$

Table 5: Held-out edge prediction, full metrics under **random** negatives, ten seeds per dataset (mean$_{\pm\text{std}}$). Same runs as Table 4; constructive embeddings use the recommended policy $P_{1.5}(r{=}2)$. The *(binary)* rows replace the slack scorer with a $0/1$ contains-vs-not indicator over the same boxes and collapse to chance: the embedding compiles only the visible 90%, so held-out positives are recoverable only from the slack. Symbolic baselines (italics) inspect the visible training graph directly. Best non-symbolic value per column in bold.

Method	AUC	AP	MRR	Hits@1	Hits@10
GO					
Constructive (ours)	$.788_{\pm.002}$	$.061_{\pm.001}$	$.304_{\pm.003}$	$.157_{\pm.003}$	$.614_{\pm.003}$
+ optimiser	$\mathbf{.796}_{\pm.002}$	$\mathbf{.097}_{\pm.003}$	$.340_{\pm.004}$	$.189_{\pm.004}$	$.642_{\pm.004}$
Constructive (binary)	$.500_{\pm.000}$	$.020_{\pm.000}$	$.039_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$
+ optimiser (binary)	$.520_{\pm.001}$	$.038_{\pm.001}$	$.076_{\pm.001}$	$.036_{\pm.001}$	$.041_{\pm.001}$
Order emb. (Vendrov et al., 2016)	$.793_{\pm.003}$	$.073_{\pm.002}$	$\mathbf{.352}_{\pm.004}$	$\mathbf{.202}_{\pm.004}$	$\mathbf{.666}_{\pm.005}$
Poincaré (Nickel & Kiela, 2017)	$.689_{\pm.004}$	$.030_{\pm.000}$	$.163_{\pm.004}$	$.043_{\pm.005}$	$.425_{\pm.007}$
Gumbel box (Dasgupta et al., 2020)	$.548_{\pm.003}$	$.023_{\pm.000}$	$.145_{\pm.003}$	$.047_{\pm.003}$	$.332_{\pm.006}$
Learned box (Vilnis et al., 2018)	$.582_{\pm.002}$	$.025_{\pm.000}$	$.138_{\pm.002}$	$.046_{\pm.002}$	$.313_{\pm.006}$
Cones (Ganea et al., 2018)	$.642_{\pm.021}$	$.031_{\pm.002}$	$.224_{\pm.013}$	$.087_{\pm.007}$	$.544_{\pm.028}$
BoxE (Abboud et al., 2020)	$.675_{\pm.003}$	$.048_{\pm.001}$	$.252_{\pm.003}$	$.117_{\pm.003}$	$.546_{\pm.007}$
Jaccard-anc.	$.831_{\pm.002}$	$.066_{\pm.001}$	$.358_{\pm.003}$	$.159_{\pm.004}$	$.725_{\pm.006}$
Adamic-Adar	$.823_{\pm.001}$	$.094_{\pm.002}$	$.297_{\pm.003}$	$.113_{\pm.002}$	$.622_{\pm.004}$
Common-anc.	$.807_{\pm.001}$	$.074_{\pm.001}$	$.289_{\pm.004}$	$.103_{\pm.002}$	$.618_{\pm.005}$
Shortest-path	$.500_{\pm.000}$	$.020_{\pm.000}$	$.038_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$
MeSH					
Constructive (ours)	$.726_{\pm.004}$	$.050_{\pm.001}$	$.220_{\pm.005}$	$.093_{\pm.006}$	$.503_{\pm.004}$
+ optimiser	$\mathbf{.738}_{\pm.004}$	$\mathbf{.054}_{\pm.002}$	$\mathbf{.237}_{\pm.006}$	$\mathbf{.104}_{\pm.006}$	$.529_{\pm.008}$
Constructive (binary)	$.500_{\pm.000}$	$.020_{\pm.000}$	$.039_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$
+ optimiser (binary)	$.503_{\pm.001}$	$.021_{\pm.000}$	$.044_{\pm.001}$	$.005_{\pm.001}$	$.005_{\pm.001}$
Order emb. (Vendrov et al., 2016)	$.696_{\pm.005}$	$.049_{\pm.001}$	$.230_{\pm.004}$	$.100_{\pm.004}$	$.516_{\pm.009}$
Poincaré (Nickel & Kiela, 2017)	$.614_{\pm.004}$	$.025_{\pm.000}$	$.119_{\pm.004}$	$.029_{\pm.003}$	$.302_{\pm.011}$
Gumbel box (Dasgupta et al., 2020)	$.572_{\pm.014}$	$.026_{\pm.002}$	$.173_{\pm.006}$	$.064_{\pm.005}$	$.413_{\pm.012}$
Learned box (Vilnis et al., 2018)	$.487_{\pm.004}$	$.018_{\pm.000}$	$.072_{\pm.003}$	$.012_{\pm.002}$	$.144_{\pm.006}$
Cones (Ganea et al., 2018)	$.672_{\pm.013}$	$.038_{\pm.001}$	$.233_{\pm.007}$	$.085_{\pm.007}$	$\mathbf{.591}_{\pm.011}$
BoxE (Abboud et al., 2020)	$.614_{\pm.005}$	$.033_{\pm.001}$	$.171_{\pm.004}$	$.065_{\pm.004}$	$.393_{\pm.007}$
Jaccard-anc.	$.761_{\pm.004}$	$.055_{\pm.001}$	$.229_{\pm.005}$	$.063_{\pm.004}$	$.565_{\pm.007}$
Adamic-Adar	$.723_{\pm.004}$	$.059_{\pm.002}$	$.188_{\pm.004}$	$.042_{\pm.003}$	$.463_{\pm.008}$
Common-anc.	$.719_{\pm.004}$	$.051_{\pm.001}$	$.186_{\pm.003}$	$.041_{\pm.003}$	$.463_{\pm.007}$
Shortest-path	$.500_{\pm.000}$	$.020_{\pm.000}$	$.038_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$
WordNet					
Constructive (ours)	$.603_{\pm.000}$	$.036_{\pm.000}$	$.160_{\pm.001}$	$.065_{\pm.002}$	$.340_{\pm.003}$
+ optimiser	$.596_{\pm.000}$	$.036_{\pm.000}$	$.161_{\pm.001}$	$.058_{\pm.002}$	$.338_{\pm.002}$
Constructive (binary)	$.500_{\pm.000}$	$.020_{\pm.000}$	$.038_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$
+ optimiser (binary)	$.500_{\pm.000}$	$.020_{\pm.000}$	$.038_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$
Order emb. (Vendrov et al., 2016)	$\mathbf{.719}_{\pm.006}$	$\mathbf{.044}_{\pm.001}$	$.217_{\pm.005}$	$.088_{\pm.003}$	$.510_{\pm.014}$
Poincaré (Nickel & Kiela, 2017)	$.571_{\pm.006}$	$.022_{\pm.000}$	$.098_{\pm.004}$	$.021_{\pm.004}$	$.226_{\pm.008}$
Gumbel box (Dasgupta et al., 2020)	$.663_{\pm.003}$	$.035_{\pm.001}$	$\mathbf{.225}_{\pm.004}$	$\mathbf{.092}_{\pm.004}$	$\mathbf{.533}_{\pm.009}$
Learned box (Vilnis et al., 2018)	$.493_{\pm.003}$	$.019_{\pm.000}$	$.075_{\pm.002}$	$.012_{\pm.001}$	$.156_{\pm.007}$
Cones (Ganea et al., 2018)	$.594_{\pm.015}$	$.028_{\pm.002}$	$.181_{\pm.009}$	$.059_{\pm.006}$	$.457_{\pm.026}$
BoxE (Abboud et al., 2020)	$.668_{\pm.007}$	$.038_{\pm.001}$	$.184_{\pm.005}$	$.070_{\pm.005}$	$.430_{\pm.015}$
Jaccard-anc.	$.754_{\pm.000}$	$.043_{\pm.000}$	$.190_{\pm.001}$	$.037_{\pm.002}$	$.483_{\pm.002}$
Adamic-Adar	$.756_{\pm.000}$	$.056_{\pm.000}$	$.177_{\pm.001}$	$.032_{\pm.001}$	$.453_{\pm.003}$
Common-anc.	$.746_{\pm.000}$	$.049_{\pm.000}$	$.177_{\pm.001}$	$.031_{\pm.001}$	$.453_{\pm.003}$
Shortest-path	$.500_{\pm.000}$	$.020_{\pm.000}$	$.038_{\pm.000}$	$.000_{\pm.000}$	$.000_{\pm.000}$

Table 6: Held-out edge prediction, full metrics under **sibling** negatives (a node sharing a direct parent with the true child, topped up from the random pool below 50 siblings), ten seeds per dataset (mean$_{\pm\text{std}}$). Same runs, methods, and conventions as Table 5. Best non-symbolic value per column in bold.

AUC and top-rank precision disagree. On MeSH the constructive slack has the higher AUC than order embeddings under both negative modes, yet under random negatives order embeddings lead AP (.225 vs. .120), MRR (.486 vs. .418), and Hits@1 (.371 vs. .295), and they lead the same metrics on GO, where AUC is tied. AUC is threshold-free and rewards ordering every positive above every negative, whereas AP and MRR reward placing the positive at the very top, and a scorer can separate positives from the bulk of the negatives well while rarely putting them first. Sibling negatives narrow

the gap: on MeSH the two methods are within .01 MRR and the optimiser variant leads every rank metric, while on GO order embeddings keep a lead of .05 MRR.

Sibling negatives erode the symbolic ceiling. Under random negatives the symbolic baselines that inspect the visible training graph sit above every geometric method on every cell. Under sibling negatives their lead collapses: Jaccard-anc. MRR drops from .763 to .358 on GO, from .549 to .229 on MeSH, and from .555 to .190 on WordNet, and its AUC drops to .831 / .761 / .754, still above every geometric method. The mechanism is structural. A sibling shares its parent with the true child, so the two have near-identical ancestor sets, and ancestor-overlap features cannot discriminate the child from a sibling of the child even with full graph access.

Constructive boxes degrade more gracefully. On MeSH and WordNet the constructive slack loses less from the switch to sibling negatives (MRR $.418 \to .220$ and $.292 \to .160$, drops of 47% and 45% against the ceiling's 58% and 66%), so the gap to the symbolic ceiling narrows to .009 on MeSH and .030 on WordNet. On GO both erode at a similar rate ($.632 \to .304$ vs. $.763 \to .358$). Jaccard-anc. stays ahead in every cell, consistent with its ceiling role in the body table.

E HYPERNYMY VERIFICATION ON FULL WORDNET (FULL TABLE)

Setup. Full WordNet noun hierarchy (82,115 synsets, 84,428 parent–child edges) embedded under $P_{1.5}(r{=}2)$. For each method we sample 2,000 pairs per relation type (hypernymy, cohyponymy, random) and score every pair under the embedding. Sub-tasks: (a) detection (hypernym vs. all negatives, AUC/AP/best-F1), (b) hard discrimination (hypernym vs. cohyponym, AUC_{hard}), and (c) direction (correct ordering of a known hypernym pair). Baselines are trained under the tuned regime of Section 7 at matched dimensionality. Ten-seed mean $\pm$ std reported across seeds $\{42, 123, 7, 100, 200, 300, 400, 500, 600, 700\}$.

Method	AUC	F1	AUC_{hard}	Dir.
Constructive (ours)	$\mathbf{1.000}_{\pm.000}$	$\mathbf{1.000}_{\pm.000}$	$\mathbf{1.000}_{\pm.000}$	$\mathbf{1.000}_{\pm.000}$
Constructive (binary)	$1.000_{\pm.000}$	$1.000_{\pm.000}$	$1.000_{\pm.000}$	$1.000_{\pm.000}$
Order emb. (Vendrov et al., 2016)	$1.000_{\pm.000}$	$.999_{\pm.000}$	$1.000_{\pm.000}$	$1.000_{\pm.001}$
Learned box (Vilnis et al., 2018)	$1.000_{\pm.000}$	$.999_{\pm.000}$	$1.000_{\pm.000}$	$1.000_{\pm.000}$
Cones (Ganea et al., 2018)	$.975_{\pm.001}$	$.907_{\pm.004}$	$.978_{\pm.002}$	$.999_{\pm.001}$
Gumbel box (Dasgupta et al., 2020)	$.983_{\pm.001}$	$.939_{\pm.003}$	$.969_{\pm.003}$	$1.000_{\pm.000}$
Poincaré (Nickel & Kiela, 2017)	$.807_{\pm.007}$	$.718_{\pm.004}$	$.769_{\pm.008}$	$.851_{\pm.007}$
BoxE (Abboud et al., 2020)	$.995_{\pm.001}$	$.960_{\pm.006}$	$.993_{\pm.002}$	$.733_{\pm.013}$
GloVe cosine	$.728_{\pm.005}$	$.582_{\pm.005}$	$.669_{\pm.006}$	$.500_{\pm.000}$
GloVe WeedsPrec	$.710_{\pm.006}$	$.562_{\pm.005}$	$.658_{\pm.009}$	$.527_{\pm.010}$
GloVe neg. distance	$.728_{\pm.005}$	$.582_{\pm.005}$	$.669_{\pm.006}$	$.500_{\pm.000}$

Table 7: Hypernymy detection on the full WordNet noun hierarchy, mean $\pm$ std over ten seeds ($x_{\pm.yyy}$ means $x \pm 0.\text{yyy}$). AUC: hypernym vs. all negatives; AUC_{hard}: hypernym vs. cohyponym hard subset; Dir.: direction accuracy on known hypernym pairs. Constructive embeddings use $P_{1.5}(r{=}2)$; positives and policy-covered cohyponym negatives are theorem-guaranteed, and both the slack and the binary scorer reach 1.000 with zero variance. Baselines are trained at matched dimensionality. The order embeddings' direction score is 0.9997 ± 0.0005 before rounding. The GloVe rows are feature baselines over pretrained vectors (Pennington et al., 2014), with each synset represented by the mean vector of the tokens of its first lemma.

Observations. *(i) Theorem-perfect on policy-covered classes, empirically perfect on the rest.* Positives and policy-covered cohyponym negatives have AUC $= 1$ exactly by Theorems 1 and 2. Random-incomparable negatives are not theorem-guaranteed (a coincidental containment in every dimension would still score 1), but the empirical rate on WordNet at the packer's output dimensionality is zero, so the full sampled benchmark also reaches AUC $= 1.000$. *(ii) Direction needs an asymmetric score.* Constructive boxes, learned boxes, order embeddings, cones, and Gumbel boxes all reach 0.999 or better on direction, because their containment scores are asymmetric by definition. Symmetric

GloVe similarity stays at the 0.500 random baseline, and the two methods whose scores are only partly directional land in between (Poincaré 0.851, BoxE 0.733).