

Compact Geometric Representations of Hierarchies

Prashant Gokhale

University of Wisconsin–Madison

Piotr Indyk

MIT

Yuhao Liu

University of Wisconsin–Madison

Sandeep Silwal

University of Wisconsin–Madison

Tony Wang

University of Wisconsin–Madison

Haike Xu

MIT

Editors: Steve Hanneke and Tor Lattimore

Abstract

Computing geometric representations of data is a cornerstone of modern machine learning, typically achieved by training dual encoders which map queries and documents into a shared embedding space. Recent work of You et al. [NeurIPS '25] has extended this approach to hierarchical retrieval, where relevance is determined by the ancestor-descendant relationships in a Directed Acyclic Graph (DAG). While previous work has shown that valid embeddings exist when the number of descendants is small, these bounds degrade significantly for deep hierarchies, requiring dimensions as large as the total number of nodes.

In this paper, we investigate compact reachability embeddings for more general graph classes and provide theoretical guarantees for representing hierarchies using embeddings whose dimension depends on structural graph parameters. We prove that for any directed tree, there exists a reachability embedding in constant dimension 3, independent of the tree’s size or depth. We generalize this result to graphs characterized by treewidth t , constructing embeddings of dimension $O(t \log n)$, where n is the number of nodes. Complementing these upper bounds, we provide matching or near-matching lower bounds, showing that dimension $\Omega(n)$ is necessary for general DAGs and $\Omega(t/\log(n/t))$ is required for graphs of treewidth t . We also obtain upper and lower bounds parameterized by the number of cross-edges in the DAG. We additionally show that our embeddings can be constructed on real world datasets, and that they give much smaller dimensions in high recall regimes compared to prior embeddings with theoretical guarantees.

Keywords: Representation Power, Embeddings, Information Retrieval, Hierarchical Retrieval, Dual Encoders

Acknowledgments

This project was supported in part by the NSF TRIPODS program (award DMS-2022448).

1. Introduction

Computing geometric representations of data, often called embeddings, is one of the key tools in modern machine learning and information retrieval. Such representations are typically computed by training a *bi-encoder*, i.e., two mappings a and b , such that a maps queries and b maps data points into some d -dimensional feature space. The mappings are designed such that the data points p that are relevant to a query q are mapped to similar representations $a(q)$ and $b(p)$, and to dissimilar representations otherwise. Formally, $a(q)$ and $b(p)$ are defined as *similar* if and only if the inner product $a(q) \cdot b(p) > r$ for some threshold r . Thanks to such embeddings, retrieval is reduced to (approximate) nearest neighbor search in a high-dimensional space, a task for which many efficient algorithms have been developed (Andoni and Indyk, 2008; Malkov and Yashunin, 2018; Jayaram Subramanya et al., 2019). This approach, often referred to as *dense retrieval*, has grown in popularity over the last two decades, exemplified by Zhan et al. (2021); Zhao et al. (2024); Fang et al. (2024); Chen et al. (2024); Wang et al. (2026); Quinn et al. (2025); Saxena et al. (2026).

The accuracy of dense retrieval is mostly determined by the quality of the embeddings. Although such quality embeddings exist for many retrieval tasks, recent work on more complex benchmarks such as BRIGHT (Su et al., 2024) demonstrates the challenges of this approach when the definition of relevance involves complex reasoning. This motivated several recent works (Guo et al., 2019; Weller et al., 2025; You et al., 2025) investigating when, and under what conditions, good bi-encoders exist.

In this paper we study embeddings for *hierarchical retrieval*, an important class of retrieval tasks where the documents are organized into an (often implicit) hierarchy that determines the relevance. For example, a document on “Poodles” is relevant to a query about a more general category of “Dogs”, while the reverse is not true in general. Such hierarchies can be modeled as Directed Acyclic Graphs (DAGs) defined over the data: a document is relevant to a query if the query corresponds to an *ancestor* of the document in the DAG. Formally, a and b are reachability embeddings if for any pair of nodes u and v in a DAG, v is reachable from u if and only if $a(u) \cdot b(v) > 0$.

A recent work (You et al., 2025) pioneered the study of embeddings that represent such notion of relevance. In a surprising finding, it shows that if the number of descendants $S(v)$ of any node v in the DAG is bounded by s , then reachability embeddings exist even in dimension of roughly $d = O(s \log n)$, where n is the number of documents.

However, their work leaves open the setting when some nodes v have many descendants. For example, if a hierarchy is a tree, then the bound s corresponds to the height of the tree, which could be as large as n . Furthermore, the aforementioned theorem in fact holds in *any* setting where the number of nodes relevant to any query q is bounded by s - the relevance sets do not need to involve reachability in a DAG (Guo et al., 2019). This raises the question of whether more fine-grained embeddings exist, especially ones that are parametrized by well-studied graph parameters.

Our results In this paper we answer the above question in the affirmative. Specifically, we present compact reachability embeddings whose dimension depends on the following graph parameters:

- *Treewidth*: We use the treewidth of the underlying undirected graph. Informally, a graph has treewidth t if it can be decomposed into balanced components by removing t nodes (see Section 2 for a formal definition). Treewidth is a well-studied graph parameter characterizing how “tree like” the graph is; in particular, the treewidth of a tree is 1. It is also a popular parameter characterizing the complexity of graph algorithms (Cygan et al., 2015), as well as algorithms

Graph Class	Upper Bound	Lower Bound
General DAG	$O(n)$ You et al. (2025)	$\Omega(n)$ Theorem 16
Treewidth t	$O(t \log n)$ Theorem 11	$\Omega\left(\frac{t}{\log \frac{n}{t}}\right)$ Theorem 20
Cross-Edges k	$3 + k$ Theorem 25	$\Omega\left(\max\left(k^{1/4}, \frac{k \log(n^2/k)}{n \log n}\right)\right)$ Theorems 29 and 30
Directed Trees	3 Theorem 9	3 Lemma 23

Table 1: A summary of our upper and lower bounds for the embedding dimension needed to represent reachability in various graph classes.

in the machine learning literature (Elidan and Gould, 2008; Scanagatta et al., 2016). In particular, many real-world graphs representing hierarchical relationships admit small treewidth; for example, for the WordNet dataset (Miller, 1994) used by You et al. (2025), a simple check on the underlying undirected graph shows that its treewidth is at most 37.

- *Cross-Edges*: A distinct measure that we also use to measure the complexity of a hierarchy is the number of cross-edges. Given any directed tree T with edges from the DAG G , an edge (u, v) is called a *cross-edge* w.r.t. T if (u, v) is an edge in G but v is not reachable from u in T . This measure is popular in the database community for parameterizing the memory of data structures answering DAG reachability queries (see Wang et al. (2006); Yildirim et al. (2010); Chen et al. (2005) and the survey Zhang et al. (2025) for details). Our work instead uses this natural parameter for characterizing the embedding dimension.

Our results are summarized in Table 1. In short, we present reachability embeddings with dimension depending linearly on parameters of interest (treewidth or the number of cross-edges). In the specific case where the graph is a directed tree, we construct embeddings into space of *constant* dimension 3. Note that in this case, prior embeddings given in You et al. (2025) required dimension as high as $\Omega(n)$, where n is the total number of nodes in the graph. We note that dimension 3 is necessary, even for directed trees of constant size (Lemma 23).

Lastly, in Appendix D, we describe two graph families, one having constant tree-width but $\Omega(n)$ number of cross-edges, and the other having $\Omega(n)$ tree-width but a constant number of cross edges, showing that our two upper bounds are incomparable in general.

2. Preliminaries

We first define reachability embeddings.

Definition 1 (Reachability) Let $G = (V, E)$ be a directed graph. We say a vertex v is reachable from another vertex u in G , denoted by $u \rightsquigarrow_G v$, if and only if there is a directed path from u to v in G . We denote by $u \not\rightsquigarrow_G v$ if there is no path from u to v in G . We may drop the subscript G when it is clear from context.

Definition 2 (Reachability Embedding) Let $G = (V, E)$ be a directed graph. A reachability embedding of dimension d in G is defined as a pair of two embeddings $a, b : V \mapsto \mathbb{R}^d$ such that for any two vertices $u, v \in V$:

$$\langle a_u, b_v \rangle > 0 \iff u \rightsquigarrow_G v.$$

Next we formally introduce treewidth and review relevant properties useful for Theorem 11. First we state the definition of a tree cover.

Definition 3 Let $G = (V, E)$ be an undirected graph, T be a tree, and $\mathcal{V} = (V_t)_{t \in T}$ be a family of vertex sets $V_t \subseteq V$ indexed by all the nodes t of T (we also call \mathcal{V} the bags of V). Then (T, \mathcal{V}) is a tree-decomposition of G if the following hold:

- $V = \cup_{t \in T} V_t$;
- For every edge $(u, v) \in E$, there exists a bag containing both u, v : that is $u, v \in V_t$ for some $t \in T$;
- For any vertex $u \in V$, all the bags containing u (i.e., $\{t \in T : u \in V_t\}$) induce a connected subtree of T .

When T is a path, the tree-decomposition is called a path-decomposition.

Intuitively, the treewidth of G is the minimum integer k such that there exists a tree decomposition of G with bags of size at most $k + 1$.

Definition 4 (Treewidth) The width of a tree decomposition of an undirected graph G is one less than the maximum bag size of that tree decomposition of G . The treewidth $\text{tw}(G)$ of G is the minimum width of all tree decompositions of G .

Treewidth has the following properties which we use in the construction of Theorem 11.

Theorem 5 (Robertson and Seymour (1986)) Let $G = (V, E)$ be an undirected graph with treewidth t . Then there exists a vertex set $S \subseteq V$ with $|S| \leq t + 1$ such that every connected component of $G[V \setminus S]$ has at most $\frac{1}{2}|V \setminus S|$ vertices. The vertex set S is also referred to as vertex separators.

Lemma 6 (Robertson and Seymour (1986)) Let $G = (V, E)$ be an undirected graph with treewidth t . Any minor H of G has treewidth at most t . Furthermore, let H be any subgraph of G . Then $\text{tw}(H) \leq \text{tw}(G) = t$.

Next we define the notion of cross-edges. This is a standard definition for the edges whose reachability relationships are not captured by a depth-first search tree of a DAG.

Definition 7 Let $G = (V, E)$ be a directed acyclic graph (DAG) and $F = (V, E_F)$ be a directed spanning-forest subgraph of G . We say $(u, v) \in E$ is a cross-edge w.r.t. F in G if $u \not\rightsquigarrow_F v$.

For simplicity, we abbreviate a cross-edge w.r.t. F in G simply as a cross-edge, where no ambiguity arises. Note that, by Definition 7, a cross-edge must be in $E - E_F$ (a non-forest edge). Additional preliminaries about standard algorithmic results relating to depth-first search and cross-edges are given in Appendix B.1.

The construction of reachability embeddings in some of our theorems is built on the reachability information of certain ‘special’ vertices (e.g. the separator vertices of Theorem 5), necessitating the following definition.

Definition 8 Let $G = (V, E)$ be a directed graph. For any vertex $v \in V$, we define the reachability sets $In(v)$ and $Out(v)$ w.r.t. G as:

$$In(v) = \{w \in V \mid w \rightsquigarrow_G v\} \cup \{v\} \quad \text{and} \quad Out(v) = \{w \in V \mid v \rightsquigarrow_G w\} \cup \{v\}.$$

3. Reachability Embeddings for Directed Trees

In this section, we study embeddings for the simplest non-trivial setting: rooted directed trees. Note a directed tree is a DAG whose underlying graph is a tree and rooted implies there is a vertex which can reach all other vertices.

Theorem 9 Let $T = (V, E)$ be a rooted directed tree. There exists a reachability embedding of dimension $d = 3$ for T .

The key observation is that in a rooted directed tree, reachability information is concisely captured by the discovery/finishing times of the standard depth-first search (DFS) algorithm (see Appendix B for an overview of DFS). Letting d_v denote the *discovery time* of a vertex v (the first time DFS visits v), and f_v denote the *finish time* (when DFS exits v), we have the following interval containment result, stated formally in Theorem 22: v is a descendant of u if and only if $[d_v, f_v] \subseteq [d_u, f_u]$.

Thus, we design vectors $a_u, b_v \in \mathbb{R}^3$ whose inner product equals $(f_u - d_v)(d_v - d_u)$, which is positive exactly when d_v lies strictly between d_u and f_u , i.e., exactly when v lies in the DFS subtree of u .

Proof [Proof of Theorem 9] Running depth-first search (DFS) on T yields discovery time d_u and the finishing time f_u for any $u \in V$, whose properties are given in Definition 21. By Theorem 22, for any $u, v \in V$, v is a descendant of u if and only if $[d_v, f_v] \subseteq [d_u, f_u]$. Thus, we can define the embeddings $a, b : V \rightarrow \mathbb{R}^3$ as follows:

$$a_u = (d_u + f_u, -1, -d_u f_u) \quad \text{and} \quad b_u = (d_u, d_u^2, 1), \quad \forall u \in V.$$

Since $\langle a_u, b_v \rangle = (d_u + f_u)d_v - d_v^2 - d_u f_u = (f_u - d_v)(d_v - d_u)$, then we have

$$\langle a_u, b_v \rangle > 0 \iff d_u < d_v < f_u \text{ or } f_u < d_v < d_u.$$

Furthermore, by Theorem 22, if $d_u < d_v < f_u$, then $[d_v, f_v] \subseteq [d_u, f_u]$. Therefore,

$$\langle a_u, b_v \rangle > 0 \iff [d_v, f_v] \subseteq [d_u, f_u] \iff v \text{ is a descendant of } u \iff u \rightsquigarrow_T v.$$

■

The construction above is component-wise: it only relies on DFS interval structure within a tree. As a result, it extends immediately to directed forests by running DFS on one tree component after another and applying the same embedding construction on each tree component since the reachability never holds across components.

Corollary 10 *Let $F = (V, E)$ be a directed forest. There exists a reachability embedding of dimension $d = 3$ in F .*

Proof Again running DFS on F yields discovery time d_u and the finishing time f_u for any $u \in V$. We still use the embeddings $a, b : V \rightarrow \mathbb{R}^3$ as described in Theorem 9. For the same reasons as in Theorem 9, we know that for any two vertices u, v in the same tree component of F , we can guarantee that:

$$\langle a_u, b_v \rangle > 0 \iff u \rightsquigarrow_F v.$$

On the other hand, for any two vertices u, v in the different tree components of F , it is clear that $u \not\rightsquigarrow_F v$. By Theorem 22, the intervals $[d_u, f_u]$ and $[d_v, f_v]$ are entirely disjoint. Thus,

$$\langle a_u, b_v \rangle = (f_u - d_v)(d_v - d_u) < 0.$$

Therefore, this pair of two embeddings is a valid reachability embedding of dimension $d = 3$ in F .

■

We note that results of [Delsarte and Kamp \(1989\)](#) and [Alon et al. \(2016\)](#) already give a construction of a directed tree which requires a reachability embedding of ≥ 3 ; see Lemma 23.

4. Reachability Embeddings Parameterized by Treewidth

This section proves our main upper bound. We note that it holds for any directed graph, not necessarily DAGs.

Theorem 11 *Let G be a directed graph with n vertices whose underlying undirected graph has treewidth t . Then G admits a reachability embedding of dimension $d(n) = O(t \log n)$.*

The construction for Theorem 11 is recursive and relies on balanced vertex separators implied by bounded treewidth (Theorem 5). At each step, we remove a separator S of size at most $t + 1$ so that the remaining graph breaks into components of size at most half. We then group these components into two sides A and B of size at most $2n/3$, embed $G[A]$ and $G[B]$ recursively. Naively, one can hope to simply combine the embeddings of the two components via concatenation, but this may create *false positives*: pairs of vertices on two different components with no reachability relationships but positive inner product. Thus, we carefully add a small number of additional coordinates to (i) eliminate false positives across the split and (ii) certify reachability that is realized via vertices in the separator.

We begin with a simple combinatorial partitioning lemma. After removing a small separator S , the graph $G[V \setminus S]$ decomposes into connected components. Then Lemma 12 will allow us to group the component sizes into two balanced sides while keeping entire components intact so that no edges cross between the two sides.

Lemma 12 *Let x_1, \dots, x_k be positive integers such that $\sum_{i=1}^k x_i = n$ and $\max_{1 \leq i \leq k} x_i \leq n/2$. Then there exists a 2-partition $[I_A, I_B]$ of $[k]$ (i.e., $I_A \cup I_B = [k]$ and $I_A \cap I_B = \emptyset$) such that:*

$$S_A = \sum_{i \in I_A} x_i \leq \frac{2n}{3} \quad \text{and} \quad S_B = \sum_{i \in I_B} x_i \leq \frac{2n}{3}.$$

Proof See Appendix B.2. ■

Next we combine Lemma 12 with the standard balanced-separator guarantee for bounded treewidth. Theorem 5 ensures that removing at most $t + 1$ vertices breaks the graph into components of size at most half. We then apply Lemma 12 to merge these components into two groups of size at most $2n/3$, obtaining a separation (A, B, S) suitable for recursion.

Lemma 13 *Let $G = (V, E)$ be an undirected graph on n vertices with treewidth t . Then there exists a separator $S \subseteq V$ with $|S| \leq t + 1$ and a partition $V \setminus S = A \sqcup B$ such that:*

1. $|A| \leq \frac{2n}{3}$ and $|B| \leq \frac{2n}{3}$,
2. there are no edges between A and B in $G[V \setminus S]$.

Proof By Theorem 5, there exists a separator $S \subseteq V$ with $|S| \leq t + 1$ such that every connected component of $G[V \setminus S]$ has size at most $|V \setminus S|/2$. Let these component sizes be x_1, \dots, x_k , so that $\sum_{i=1}^k x_i = m$ where $m := |V \setminus S|$, and $\max_i x_i \leq m/2$. Applying Lemma 12, we can partition the components into two groups whose total sizes are each at most $2m/3$. Let A (resp. B) be the union of components in the first (resp. second) group. Then $|A| \leq 2m/3$ and $|B| \leq 2m/3$, and there are no edges between A and B in $G[V \setminus S]$ because they are unions of distinct connected components. Finally, since $m \leq n$, we also have $|A|, |B| \leq 2n/3$. ■

To prove Theorem 11, given a balanced separation $V \setminus S = A \sqcup B$, we embed $G[A]$ and $G[B]$ recursively. We then add two separation coordinates that impose a large negative contribution on cross-pairs (A, B) , preventing false positives across the split. Finally, we add $|S|$ connection coordinates to contribute a large positive term exactly when a directed path from u to v can be realized via some separator vertex in S .

Proof [Proof of Theorem 11] We prove the claim by induction on $n = |V|$. The base case $n = 1$ is immediate. Let $f(n)$ be the number of dimensions required for n . Assume the theorem holds for all directed graphs on fewer than n vertices whose underlying undirected graph has treewidth at most t . Let $G = (V, E)$ be a directed graph on n vertices, and suppose the underlying undirected graph has treewidth t . By Lemma 13, there exists a separator $S = \{s_1, \dots, s_k\}$ with $k \leq t + 1$ and a partition $V \setminus S = A \sqcup B$ such that $|A|, |B| \leq 2n/3$ and there are no edges between A and B in $G[V \setminus S]$. By Lemma 6, the induced subgraphs $G[A]$ and $G[B]$ have treewidth at most t . By the inductive hypothesis, $G[A]$ and $G[B]$ admit reachability embeddings (a^A, b^A) and (a^B, b^B) of dimension $f(2n/3)$. We construct embeddings $a, b : V \mapsto \mathbb{R}^{f(2n/3)+2+k}$ by concatenating three blocks of coordinates.

Coordinate construction.

1. **Recursion Coordinates.** Define $a^{\text{rec}}, b^{\text{rec}} : V \mapsto \mathbb{R}^{f(2n/3)}$ by

$$a_u^{\text{rec}} = \begin{cases} a_u^A & u \in A, \\ a_u^B & u \in B, \\ \mathbf{0} & u \in S, \end{cases} \quad b_u^{\text{rec}} = \begin{cases} b_u^A & u \in A, \\ b_u^B & u \in B, \\ \mathbf{0} & u \in S. \end{cases}$$

In particular, for $u, v \in A$ we have $\langle a_u^{\text{rec}}, b_v^{\text{rec}} \rangle > 0 \iff u \rightsquigarrow_{G[A]} v$, and similarly for $u, v \in B$. If $u \in S$ or $v \in S$ then $\langle a_u^{\text{rec}}, b_v^{\text{rec}} \rangle = 0$.

2. **Separation Coordinates.** Let $M_1 > 0$ satisfy

$$M_1 > \max_{p, q \in V} \langle a_p^{\text{rec}}, b_q^{\text{rec}} \rangle.$$

Define $a^{\text{sep}}, b^{\text{sep}} : V \mapsto \mathbb{R}^2$ by

$$a_u^{\text{sep}} = \begin{cases} (1, 0) & u \in A, \\ (0, 1) & u \in B, \\ (0, 0) & u \in S, \end{cases} \quad b_u^{\text{sep}} = \begin{cases} (0, -M_1) & u \in A, \\ (-M_1, 0) & u \in B, \\ (0, 0) & u \in S. \end{cases}$$

Then for any $u, v \in V$,

$$\langle a_u^{\text{sep}}, b_v^{\text{sep}} \rangle = \begin{cases} -M_1 & \text{if } (u \in A \text{ and } v \in B) \text{ or } (u \in B \text{ and } v \in A), \\ 0 & \text{otherwise.} \end{cases}$$

3. **Connection Coordinates.** Let $M_2 > 0$ satisfy

$$M_2^2 > M_1 - \min_{p, q \in V} \langle a_p^{\text{rec}}, b_q^{\text{rec}} \rangle.$$

Define $a^{\text{conn}}, b^{\text{conn}} : V \mapsto \mathbb{R}^k$ by, for each $i \in [k]$,

$$a_u^{\text{conn}}[i] = \begin{cases} M_2 & \text{if } u \rightsquigarrow_G s_i, \\ 0 & \text{otherwise,} \end{cases} \quad b_u^{\text{conn}}[i] = \begin{cases} M_2 & \text{if } s_i \rightsquigarrow_G u, \\ 0 & \text{otherwise.} \end{cases}$$

Thus $\langle a_u^{\text{conn}}, b_v^{\text{conn}} \rangle > 0$ iff there exists $s_i \in S$ with $u \rightsquigarrow_G s_i$ and $s_i \rightsquigarrow_G v$.

Finally, we concatenate the three coordinates and define the reachability embeddings: for all $u \in V$,

$$a_u := (a_u^{\text{rec}}; a_u^{\text{sep}}; a_u^{\text{conn}}), \quad b_u := (b_u^{\text{rec}}; b_u^{\text{sep}}; b_u^{\text{conn}}),$$

so that

$$\langle a_u, b_v \rangle = \langle a_u^{\text{rec}}, b_v^{\text{rec}} \rangle + \langle a_u^{\text{sep}}, b_v^{\text{sep}} \rangle + \langle a_u^{\text{conn}}, b_v^{\text{conn}} \rangle.$$

Correctness. We prove that $\langle a_u, b_v \rangle > 0$ if and only if $u \rightsquigarrow_G v$. Call a pair (u, v) *reachable* $u \rightsquigarrow_G v$ via S if there exists a separator vertex $s_i \in S$ with $u \rightsquigarrow_G s_i$ and $s_i \rightsquigarrow_G v$. Every directed $u \rightarrow v$ path either passes through S (yielding reachability via S) or avoids S altogether. In the latter case the path is contained in $G[V \setminus S]$; because there are no edges between A and B in this subgraph, we must have u and v on the same side, i.e., $u \rightsquigarrow_{G[A]} v$ when $u, v \in A$ or $u \rightsquigarrow_{G[B]} v$ when $u, v \in B$.

- **Case 1:** $u \rightsquigarrow_G v$ via S . Then $\langle a_u^{\text{conn}}, b_v^{\text{conn}} \rangle \geq M_2^2$. Moreover, $\langle a_u^{\text{sep}}, b_v^{\text{sep}} \rangle \geq -M_1$ always, even if $u, v \in A$ (resp. $u, v \in B$). Hence, the inner product of their reachability embeddings:

$$\begin{aligned} \langle a_u, b_v \rangle &= \langle a_u^{\text{rec}}, b_v^{\text{rec}} \rangle + \langle a_u^{\text{sep}}, b_v^{\text{sep}} \rangle + \langle a_u^{\text{conn}}, b_v^{\text{conn}} \rangle \\ &\geq \min_{p,q \in V} \langle a_p^{\text{rec}}, b_q^{\text{rec}} \rangle - M_1 + M_2^2 > 0 \end{aligned}$$

by the choice of M_2 .

- **Case 2:** $u \rightsquigarrow_G v$ not via S . Then any $u \rightarrow v$ path lies entirely in $V \setminus S$. Since there are no edges between A and B in $G[V \setminus S]$, we must have either $u, v \in A$ or $u, v \in B$. Assume $u, v \in A$ (the other case is identical). Then $u \rightsquigarrow_{G[A]} v$, so $\langle a_u^{\text{rec}}, b_v^{\text{rec}} \rangle > 0$ by the inductive embedding on $G[A]$, and $\langle a_u^{\text{sep}}, b_v^{\text{sep}} \rangle = 0$. Also $\langle a_u^{\text{conn}}, b_v^{\text{conn}} \rangle = 0$ by assumption. Therefore $\langle a_u, b_v \rangle > 0$.
- **Case 3:** $u \not\rightsquigarrow_G v$. Then in particular u cannot reach v via any separator vertex, so $\langle a_u^{\text{conn}}, b_v^{\text{conn}} \rangle = 0$. We show $\langle a_u^{\text{rec}}, b_v^{\text{rec}} \rangle + \langle a_u^{\text{sep}}, b_v^{\text{sep}} \rangle \leq 0$.

(i) u and v lie on different sides. If $u \in A, v \in B$ or $u \in B, v \in A$, then $\langle a_u^{\text{sep}}, b_v^{\text{sep}} \rangle = -M_1$, and hence

$$\langle a_u, b_v \rangle \leq \max_{p,q \in V} \langle a_p^{\text{rec}}, b_q^{\text{rec}} \rangle - M_1 < 0$$

by the choice of M_1 .

(ii) u and v lie on the same side. Assume $u, v \in A$ (the case $u, v \in B$ is identical). Then $\langle a_u^{\text{sep}}, b_v^{\text{sep}} \rangle = 0$. Since $u \not\rightsquigarrow_G v$, we also have $u \not\rightsquigarrow_{G[A]} v$, and therefore $\langle a_u^{\text{rec}}, b_v^{\text{rec}} \rangle \leq 0$ by correctness of the recursive embedding on $G[A]$. Thus $\langle a_u, b_v \rangle \leq 0$.

(iii) at least one vertex lies in S . If $u \in S$ then $a_u^{\text{rec}} = a_u^{\text{sep}} = \mathbf{0}$, and if $v \in S$ then $b_v^{\text{rec}} = b_v^{\text{sep}} = \mathbf{0}$. Together with $\langle a_u^{\text{conn}}, b_v^{\text{conn}} \rangle = 0$, this implies $\langle a_u, b_v \rangle = 0$.

Dimension bound. The construction uses $f(2n/3)$ recursive coordinates, 2 separation coordinates, and $k \leq t + 1$ connection coordinates, hence

$$f(n) \leq f(2n/3) + 2 + (t + 1) = f(2n/3) + O(t).$$

Unrolling the recurrence yields $f(n) = O(t \log n)$, completing the proof. ■

5. Lower Bound on Embedding Dimension

This section establishes lower bounds on the dimension required by reachability embeddings. In the main body we present our lower bounds for general DAGs (Theorem 16), as well as our lower bound for graphs with treewidth t (Theorem 20). Our lower bound parameterized by the number of cross-edges is deferred to Appendix E.

We start with a general lower bound for arbitrary DAGs, and then refine it for graph families with bounded treewidth. Following Weller et al. (2025), our main tool is a reduction from embedding dimension to the *sign rank* of the reachability matrix, which allows us to invoke known counting bounds for low sign-rank sign matrices.

Definition 14 (Sign rank) For $x \in \mathbb{R}$, define $\text{sign}(x) = +1$ if $x > 0$ and $\text{sign}(x) = -1$ otherwise. For a matrix $M \in \{\pm 1\}^{n \times n}$, its *sign rank*, denoted by $\text{rank}_{\pm}(M)$, is the minimum rank of a real matrix $R \in \mathbb{R}^{n \times n}$ such that $M_{ij} = \text{sign}(R_{ij})$ for all i, j .

Suppose $G = (V, E)$ with $|V| = n$ admits a reachability embedding in dimension d , given by maps $a, b : V \mapsto \mathbb{R}^d$ satisfying the convention

$$\langle a_u, b_v \rangle > 0 \iff u \rightsquigarrow_G v, \quad (1)$$

and thus $\langle a_u, b_v \rangle \leq 0$ whenever $u \not\rightsquigarrow_G v$. Stacking the vectors into matrices $A, B \in \mathbb{R}^{n \times d}$ and defining $R := AB^{\top}$, we obtain an $n \times n$ real matrix whose entry R_{uv} equals $\langle a_u, b_v \rangle$. Let $M \in \{\pm 1\}^{n \times n}$ be the induced reachability sign matrix $M_{uv} := \text{sign}(R_{uv})$. By (1), $M_{uv} = +1$ if and only if $u \rightsquigarrow_G v$. Crucially, the factorization $R = AB^{\top}$ implies $\text{rank}(R) \leq d$ because both A and B have at most d columns. Therefore the reachability sign matrix M is realized by a rank- $\leq d$ real matrix, and hence

$$\text{rank}_{\pm}(M) \leq \text{rank}(R) \leq d.$$

Any lower bound on $\text{rank}_{\pm}(M)$ immediately yields a lower bound on the embedding dimension d . We also rely on the following counting bound on the number of matrices with low sign rank.

Lemma 15 (Lemma 24 of Alon et al. (2016), Lemma 1.1 of Hatami et al. (2022)) Let $d \leq n/2$. The number of sign matrices $M \in \{\pm 1\}^{n \times n}$ with $\text{rank}_{\pm}(M) \leq d$ is at most $(O(n/d))^{2dn} \leq 2^{O(dn \log \frac{n}{d})} \leq 2^{O(dn \log n)}$.

5.1. General Lower Bound

The next theorem follows from a counting argument: (1) Lemma 15 upper bounds how many reachability sign matrices can be represented in dimension d , while (2) an explicit family of DAGs yields exponentially many distinct reachability sign matrices.

Theorem 16 (Combinatorial Lower Bound) Any reachability embedding scheme for DAGs on n vertices requires dimension $d = \Omega(n)$ in the worst case.

Proof Let d be the embedding dimension. Fix $d \leq n/2$ and let \mathcal{M}_d be the set of $n \times n$ sign matrices that arise as the sign pattern $\text{sign}(AB^{\top})$ of some embedding $a, b : V \mapsto \mathbb{R}^d$ (equivalently, matrices of sign rank at most d). By Lemma 15, we have

$$|\mathcal{M}_d| \leq 2^{O(dn \log \frac{n}{d})}. \quad (2)$$

We now lower bound the number of distinct reachability matrices that can occur for DAGs on n vertices by an explicit family. Assume for simplicity that n is even and partition the vertex set into two parts L and R with $|L| = |R| = n/2$. Consider directed bipartite graphs G that only have edges from L to R (and no edges inside L or inside R). Such a graph is a DAG, and moreover every directed path has length at most 1. Hence for $u \in L$ and $v \in R$ we have

$$u \rightsquigarrow_G v \iff (u, v) \in E.$$

Therefore, distinct edge sets $E \subseteq L \times R$ yield distinct reachability matrices. There are $|L \times R| = (n/2)^2 = n^2/4$ potential edges from L to R , each chosen independently, so this family contains $2^{n^2/4}$ distinct reachability matrices. In particular, any class \mathcal{M}_d that represents all reachability matrices of DAGs on n vertices must satisfy

$$|\mathcal{M}_d| \geq 2^{n^2/4}. \quad (3)$$

Combining (2) and (3) gives $2^{O(dn \log \frac{n}{d})} \geq 2^{n^2/4}$, and taking logarithms yields $O(dn \log \frac{n}{d}) \geq \Omega(n^2)$. Rearranging, and solving for d proves $d = \Omega(n)$. ■

The same asymptotic lower bound also follows from classical counting results for partial orders. In particular, Kleitman and Rothschild (1970, Theorem 1) show that the number of labeled partial orders on n elements is $2^{n^2/4 + o(n^2)}$. Since reachability in a DAG is a partial order, this yields a corresponding lower bound on the number of reachability sign matrices. Plugging this stronger counting bound into the sign-rank argument above again gives $d = \Omega(n)$.

5.2. Treewidth-Parameterized Lower Bound

We now consider the case where the input graphs have bounded treewidth. We construct a specific family of graphs with treewidth t that generates a large number of distinct reachability matrices.

Definition 17 *Let n and t be integers with $1 \leq t < n$. We define $\mathcal{G}_{n,t}$ to be the family of directed graphs on vertex set $V = \{1, \dots, n\}$ such that a graph $G = (V, E)$ is in $\mathcal{G}_{n,t}$ if and only if every edge $(u, v) \in E$ satisfies: (1) u is odd and v is even, (2) $u < v$, and (3) $v - u \leq t$.*

We also state the definition of pathwidth which will be useful in lower bounding the treewidth of graphs in $\mathcal{G}_{n,t}$.

Definition 18 (Pathwidth) *The width of a path decomposition of an undirected graph G is one less than the maximum bag size of that path decomposition of G . The pathwidth $\text{pw}(G)$ of G is the minimum width of all path decompositions of G .*

Lemma 19 *Any graph $G \in \mathcal{G}_{n,t}$ has treewidth at most t .*

Proof We show that G has pathwidth at most t . Consider the sequence of subsets X_1, \dots, X_{n-t} where $X_i = \{i, i+1, \dots, i+t\}$. For any edge $(u, v) \in E$, we have $u < v \leq u+t$, so both endpoints are contained in the set X_u . Furthermore, for any vertex v , the indices i such that $v \in X_i$ form the interval $[\max(1, v-t), v]$, which is contiguous. Since each set has size $t+1$, the pathwidth (and thus treewidth) is at most $(t+1) - 1 = t$. ■

Theorem 20 *For the embedding problem on DAGs with treewidth t , the required dimension is $d = \Omega\left(t/\log\frac{n}{t}\right)$.*

Proof Let d be the embedding dimension. Consider the graphs in $\mathcal{G}_{n,t}$. Since edges only go from odd vertices to even vertices, the graph is bipartite and maximum path length is 1. Therefore, for all distinct edge sets, the corresponding reachability relations are distinct.

We now count the number of possible graphs in $\mathcal{G}_{n,t}$. For each odd vertex u , there are approximately $t/2$ valid even vertices v such that $u < v \leq u + t$. Since there are $n/2$ odd vertices, the total number of possible edges is roughly $(n/2) \cdot (t/2) = nt/4$. Each potential edge can be chosen independently, so the number of distinct graphs in $\mathcal{G}_{n,t}$ is at least $2^{nt/4}$. Using Lemma 15, the number of sign matrices of rank d is at most $2^{O(dn \log \frac{n}{d})}$. Rearranging, and solving for d we get,

$$2^{O(dn \log \frac{n}{d})} \geq 2^{nt/4} \implies O\left(dn \log \frac{n}{d}\right) \geq \Omega(nt) \implies d \geq \Omega\left(\frac{t}{\log \frac{n}{t}}\right).$$

■

6. Experiments

We empirically evaluate the representational capacity of our construction based on balanced vertex separators (Section 4). By demonstrating that our algorithm achieves perfect retrieval recall on large-scale real-world hierarchies using low dimensions, we validate our theoretical analysis. These results support our $O(t \log n)$ bound and indicate that the constant factors involved are modest.

We compare our approach against the ‘‘Handcrafted Embeddings’’ (which essentially applies JL lemma under the hood) in (You et al., 2025). While (You et al., 2025) also demonstrate a Learned Embedding which optimizes query-document similarity via stochastic gradient descent to achieve high performance in low dimensions, it does not guarantee exact retrieval, requires computationally expensive training, and has no theoretical guarantees. In contrast, our method is faster, constructive, deterministic, and ensures 100% recall. Furthermore, it is supported by a theoretical guarantee for low treewidth graphs.

6.1. Implementation Details

We implement the deterministic construction defined in Section 4, which builds embeddings recursively using balanced vertex separators. To approximate balanced vertex separators, we utilize the KaHIP library (Sanders and Schulz, 2013). We ensure numerical stability in deep hierarchies by stepwise renormalization of embeddings.

Setup We evaluate on three hierarchical datasets: *WordNet* ($N = 82\text{k}$, DAG) (Miller, 1994), *Gene Ontology* ($N = 24\text{k}$, DAG) (Ashburner et al., 2000; Consortium, 2025), and *Cora* ($N = 2.7\text{k}$, citation graph — has some cycles) (McCallum, 2017). The task is to retrieve all ancestors for a query. We compare against the handcrafted embedding of (You et al., 2025), which sums normalized ancestor vectors. Following (You et al., 2025), we report *Recall@k*. We display the baseline’s performance across a range of dimensions to illustrate its scaling behavior, highlighting the dimension required to attain $> 95\%$ recall, consistent with the success threshold established in their work.

6.2. Results

Table 2 compares our embedding against the handcrafted embedding of You et al. (2025).

Table 2: Comparison of Recall@k (%). Bold values denote recall > 95%.

Dataset	Ours Dim (100% Recall@k)	Baseline Method	Recall@k by Dimension				
			64	128	256	512	1024
WordNet	152	Handcrafted You et al. (2025)	14.4	50.9	86.1	99.2	100.0
Cora	192	Handcrafted You et al. (2025)	69.5	82.0	91.2	95.4	96.9
Gene Ontology	492	Handcrafted You et al. (2025)	22.6	53.1	82.9	96.8	99.6

Our method consistently achieves *100% Recall* with only a modest number of dimensions. On WordNet, our embedding solves the retrieval task exactly with $d = 152$. In comparison, the handcrafted baseline yields only 50.9% recall at a similar dimension ($d = 128$) and requires $d = 512$ to cross the 95% success threshold—a $3.4\times$ increase in dimensionality. While the Learned (Pretrain-Finetune) baseline You et al. (2025) performs significantly better than the handcrafted version (achieving 92.3% recall at $d = 64$ on WordNet), it still does not guarantee exact recall. Similarly, on Cora and Gene Ontology, our method guarantees exactness at dimensions lower than those required by the handcrafted baseline. This empirically validates our theoretical bound and demonstrates the efficiency of our constructive approach. Furthermore, our embedding would improve with access to a better vertex separator oracle.

References

- Rakesh Agrawal, Alexander Borgida, and Hosagrahar Visvesvaraya Jagadish. Efficient management of transitive relationships in large data and knowledge bases. *ACM SIGMOD Record*, 18(2):253–262, 1989.
- Noga Alon, Shay Moran, and Amir Yehudayoff. Sign rank versus VC dimension. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 47–80, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL <https://proceedings.mlr.press/v49/alon16.html>.
- Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008.
- M Ashburner, C A Ball, J A Blake, D Botstein, H Butler, J M Cherry, A P Davis, K Dolinski, S S Dwight, J T Eppig, M A Harris, D P Hill, L Issel-Tarver, A Kasarskis, S Lewis, J C Matese, J E Richardson, M Ringwald, G M Rubin, and G Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat. Genet.*, 25(1):25–29, May 2000.
- Hans L Bodlaender. A tourist guide through treewidth. *Developments in Theoretical Computer Science*, 6:1, 1994.

- Li Chen, Amarnath Gupta, and M Erdem Kurul. Stack-based algorithms for pattern matching on dags. In *Proceedings of the 31st international conference on Very large data bases*, pages 493–504, 2005.
- Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. Dense x retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177, 2024.
- The Gene Ontology Consortium. The gene ontology knowledgebase in 2026. *Nucleic Acids Research*, 54(D1):D1779–D1792, 12 2025. ISSN 1362-4962. doi: 10.1093/nar/gkaf1292. URL <https://doi.org/10.1093/nar/gkaf1292>.
- Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.
- Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 5. Springer, 2015.
- Philippe Delsarte and Yves Kamp. Low rank matrices with a given sign pattern. *SIAM journal on discrete mathematics*, 2(1):51–63, 1989.
- Gal Elidan and Stephen Gould. Learning bounded treewidth bayesian networks. *Advances in neural information processing systems*, 21, 2008.
- Yan Fang, Jingtao Zhan, Qingyao Ai, Jiabin Mao, Weihang Su, Jia Chen, and Yiqun Liu. Scaling laws for dense retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1339–1349, 2024.
- Jürgen Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *Journal of Computer and System Sciences*, 65(4):612–625, 2002.
- Chuan Guo, Ali Mousavi, Xiang Wu, Daniel N Holtmann-Rice, Satyen Kale, Sashank Reddi, and Sanjiv Kumar. Breaking the glass ceiling for embedding-based classifiers for large output spaces. *Advances in Neural Information Processing Systems*, 32, 2019.
- Hamed Hatami, Pooya Hatami, William Pires, Ran Tao, and Rosie Zhao. Lower Bound Methods for Sign-Rank and Their Limitations. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)*, volume 245 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:24, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-249-5. doi: 10.4230/LIPIcs.APPROX/RANDOM.2022.22. URL <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.APPROX/RANDOM.2022.22>.
- Kathy J Horadam. *Hadamard matrices and their applications*. Princeton university press, 2012.

- Suhas Jayaram Subramanya, Fnu Devvrit, Harsha Vardhan Simhadri, Ravishankar Krishnawamy, and Rohan Kadekodi. Diskann: Fast accurate billion-point nearest neighbor search on a single node. *Advances in Neural Information Processing Systems*, 32, 2019.
- Ruoming Jin, Ning Ruan, Yang Xiang, and Haixun Wang. Path-tree: An efficient reachability indexing scheme for large directed graphs. *ACM Transactions on Database Systems (TODS)*, 36(1):1–44, 2011.
- Daniel Kleitman and Bruce Rothschild. The number of finite topologies. *Proceedings of the American mathematical Society*, 25(2):276–282, 1970.
- Tuukka Korhonen and Daniel Lokshantov. An improved parameterized algorithm for treewidth. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 528–541, 2023.
- Yu A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836, 2018.
- Andrew McCallum. Cora dataset, 2017.
- George A. Miller. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. URL <https://aclanthology.org/H94-1111/>.
- Derrick Quinn, E. Ezgi Yücel, Martin Prammer, Zhenxing Fan, Kevin Skadron, Jignesh M. Patel, José F. Martínez, and Mohammad Alian. Drex: Accurate and scalable dense retrieval acceleration via algorithmic-hardware codesign. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture, ISCA '25*, page 1108–1124, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400712616. doi: 10.1145/3695053.3731079. URL <https://doi.org/10.1145/3695053.3731079>.
- Neil Robertson and Paul D. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986.
- Peter Sanders and Christian Schulz. Think locally, act globally: Highly balanced graph partitioning. In Vincenzo Bonifaci, Camil Demetrescu, and Alberto Marchetti-Spaccamela, editors, *Experimental Algorithms*, pages 164–175, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-38527-8.
- Yash Saxena, Ankur Padia, Kalpa Gunaratna, and Manas Gaur. Imrnns: An efficient method for interpretable dense retrieval via embedding modulation, 2026. URL <https://arxiv.org/abs/2601.20084>.
- Mauro Scanagatta, Giorgio Corani, Cassio P De Campos, and Marco Zaffalon. Learning treewidth-bounded bayesian networks with thousands of variables. *Advances in neural information processing systems*, 29, 2016.
- Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S Siegel, Michael Tang, et al. Bright: A realistic and challenging benchmark for reasoning-intensive retrieval. *arXiv preprint arXiv:2407.12883*, 2024.

Haixun Wang, Hao He, Jun Yang, Philip S Yu, and Jeffrey Xu Yu. Dual labeling: Answering graph reachability queries in constant time. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 75–75. IEEE, 2006.

Zihao Wang, Hang Yin, Lihui Liu, Hanghang Tong, Yangqiu Song, Ginny Wong, and Simon See. \mathbb{R}^{2k} is theoretically large enough for embedding-based top- k retrieval, 2026. URL <https://arxiv.org/abs/2601.20844>.

Orion Weller, Michael Boratko, Iftexhar Naim, and Jinhyuk Lee. On the theoretical limitations of embedding-based retrieval. *arXiv preprint arXiv:2508.21038*, 2025.

Hilmi Yildirim, Vineet Chaoji, and Mohammed J Zaki. Grail: Scalable reachability index for large graphs. *Proceedings of the VLDB Endowment*, 3(1-2):276–284, 2010.

Chong You, Rajesh Jayaram, Ananda Theertha Suresh, Robin Nittka, Felix X. Yu, and Sanjiv Kumar. Hierarchical retrieval: The geometry and a pretrain-finetune recipe. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=ktk4mxNzFe>.

Jingtao Zhan, Jiabin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 1503–1512, 2021.

Chao Zhang, Angela Bonifati, and M Tamer Özsu. Indexing techniques for graph reachability queries. *ACM Computing Surveys*, 58(6):1–35, 2025.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60, 2024.

Appendix A. Other Related Works

Related Works on Graph Parameters. The notion of treewidth was introduced by [Robertson and Seymour \(1986\)](#), as a central tool in analyzing graph structure and designing graph algorithms. A long line of work shows that NP-hard problems can be solved in polynomial time on classes of graphs with bounded treewidth (we refer to the textbook ([Cygan et al., 2015](#)), the survey ([Bodlaender, 1994](#)) or a detailed overview of the state of the art ([Korhonen and Lokshtanov, 2023](#)) for details).

The problem of efficiently determining reachability in Directed Acyclic Graphs (DAGs) has been extensively studied in the database community. In particular, the “Tree Cover” methodology starts with a spanning tree and then resolves the remaining reachability information from cross-edges using techniques such as building a look up table; see the works of [Agrawal et al. \(1989\)](#); [Chen et al. \(2005\)](#); [Wang et al. \(2006\)](#); [Jin et al. \(2011\)](#); [Zhang et al. \(2025\)](#) for more details. This line of work aims to capture reachability relationships in data structures, with the trade-offs of data structure construction time, query time, and data structure space being studied. In contrast, our work studies embedding reachability structures in low dimensional space.

Appendix B. Additional Preliminaries

B.1. Useful Facts About Depth-First Search

We recall the following standard facts about depth-first search (DFS), e.g. see the textbook of [Cormen et al. \(2022\)](#). This underlies the reachability embedding for directed trees in Appendix 3.

Proposition 21 *Consider the standard DFS procedure on a directed graph $G = (V, E)$. We have the following observations:*

1. *If G is connected, the depth-first forest consists of exactly one depth-first tree.*
2. *When DFS on G is finished, each vertex $u \in V$ has been assigned a discovery time d_u (the first time DFS visits u) and a finishing time f_u (the time when DFS exists u).*
3. *All timestamps (discovery and finishing times) are distinct integers.*

It is easy to see (and a classical fact) that the discovery and finishing times have the following properties.

Theorem 22 (Parenthesis theorem) *In any DFS of a directed graph $G = (V, E)$, for any two vertices u and v , exactly one of the following holds:*

- *the intervals $[d_u, f_u]$ and $[d_v, f_v]$ are entirely disjoint, and neither u nor v is a descendant of the other in the depth-first forest,*
- *the interval $[d_u, f_u]$ is contained entirely within the interval $[d_v, f_v]$, and u is a descendant of v in a depth-first tree, or*
- *the interval $[d_v, f_v]$ is contained entirely within the interval $[d_u, f_u]$, and v is a descendant of u in a depth-first tree.*

We also note that prior work implies three dimensions are necessary for a particularly simple directed tree.

Lemma 23 *There exists a directed tree G on $O(1)$ nodes which does not admit a reachability embedding of dimension ≤ 2 .*

Proof It is shown in [Delsarte and Kamp \(1989\)](#) and [Alon et al. \(2016\)](#) that for sufficiently large constant m , the $m \times m$ sign matrix with diagonals equal to 1 and off-diagonals equal to -1 has sign rank 3. This sign-matrix is a sub-matrix of the induced reachability sign matrix of the star-graph. The claim follows from noting that the sign-rank is a lower bound on the reachability embedding dimension (see the discussion in Section 5). ■

B.2. Useful combinatorial partitioning lemma

Lemma 24 *Let x_1, \dots, x_k be positive integers such that $\sum_{i=1}^k x_i = n$ and $\max_{1 \leq i \leq k} x_i \leq n/2$. Then there exists a 2-partition $[I_A, I_B]$ of $[k]$ (i.e., $I_A \cup I_B = [k]$ and $I_A \cap I_B = \emptyset$) such that:*

$$S_A = \sum_{i \in I_A} x_i \leq \frac{2n}{3} \quad \text{and} \quad S_B = \sum_{i \in I_B} x_i \leq \frac{2n}{3}.$$

Proof [Proof of Lemma 12] We consider two cases:

- **Case 1: If $x_i < n/3$ holds for all $i \in [k]$.** Reorder so that x_1, \dots, x_k are in non-increasing order. Let j be the largest index such that $\sum_{i=1}^j x_i \leq 2n/3$ so that $\sum_{i=1}^{j+1} x_i > 2n/3$. Set $I_A = \{1, \dots, j\}$ and $I_B = \{j+1, \dots, k\}$. Then

$$S_A = \sum_{i=1}^j x_i \leq 2n/3$$

and

$$S_B = n - S_A \geq n - 2n/3 = n/3.$$

Since $x_{j+1} < n/3$ we have

$$S_B = x_{j+1} + \sum_{i=j+2}^k x_i < n/3 + (n - \sum_{i=1}^{j+1} x_i) < n/3 + n/3 = 2n/3.$$

- **Case 2: If exists i with $x_i \geq n/3$.** Let $x_1 \geq n/3$. Set $I_A = \{1\}$ and $I_B = \{2, \dots, k\}$. Then $S_A = x_1 \leq n/2$ and $S_B = n - x_1 \leq 2n/3$. ■

Appendix C. Reachability Embeddings for DAGs Parameterized by Cross-Edges

In Section 3 we constructed a 3-dimensional embedding that exactly captures reachability in a directed forest. For a general DAG, the obstruction is that a spanning forest $F \subseteq G$ may miss directed paths that rely on non-forest edges (i.e., cross-edges). In this section we show a simple augmentation principle: each cross-edge that creates a mismatch between reachability in F and in G can be “repaired” by introducing one additional coordinate. As a consequence, we obtain an embedding whose dimension grows linearly with the number of cross-edges.

Informally, the forest embedding certifies reachability whenever a path exists entirely inside F . For any remaining reachable pair (u, v) , every directed path from u to v in G must traverse two disjoint subtrees of T , thus contain at least one cross-edge. We use the coordinate corresponding to that cross-edge to contribute a large positive term M^2 , dominating any negative contribution from the forest part.

Theorem 25 *Let $G = (V, E)$ be a DAG with a spanning forest F . Denote by E_{cross} the set of all cross-edges, and let $k = |E_{\text{cross}}|$. Then there exists a reachability embedding of G with dimension $d = 3 + k$.*

We start from the 3-dimensional forest embedding and add one coordinate per cross-edge. The new coordinate for each cross-edge (x_i, y_i) activates exactly for vertices that can reach into x_i and vertices reachable from y_i . Choosing M sufficiently large ensures that any reachable pair not already captured by the forest gains a positive M^2 contribution, while unreachable pairs cannot simultaneously activate any coordinate.

Proof [Proof of Theorem 25] By Corollary 10, there exists a reachability embedding of dimension $d' = 3$ in F , denoted by a^F and b^F . By the definition of reachability embedding, we have that for any two vertices $u, v \in V$:

$$\langle a_u^F, b_v^F \rangle > 0 \iff u \rightsquigarrow_F v.$$

Suppose that $E_{\text{cross}} = \{e_1, \dots, e_k\}$, where $e_i = (x_i, y_i)$. We consider the reachability sets $\text{In}(x_i)$ and $\text{Out}(y_i)$ w.r.t. G for any $i \in [k]$. Let M be a sufficiently large positive constant such that

$$M^2 + \min_{u, v \in V} \langle a_u^F, b_v^F \rangle > 0.$$

Then we define the embeddings $a^{\text{cross}}, b^{\text{cross}} : V \mapsto \mathbb{R}^k$ as follows:

$$a_u^{\text{cross}}[i] = \begin{cases} M & \text{if } u \in \text{In}(x_i) \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad b_u^{\text{cross}}[i] = \begin{cases} M & \text{if } u \in \text{Out}(y_i) \\ 0 & \text{otherwise} \end{cases}, \quad \forall u \in V, i \in [k],$$

where $a_u^{\text{cross}}[i]$ and $b_u^{\text{cross}}[i]$ denote the i -th coordinates of a_u^{cross} and b_u^{cross} respectively.

Finally, we define the embedding $a : V \mapsto \mathbb{R}^{3+k}$ as a concatenation of a^F and a^{cross} :

$$a_u = (a_u^F; a_u^{\text{cross}}), \quad \text{for all } u \in V.$$

Similarly, we define the embedding $b : V \mapsto \mathbb{R}^{3+k}$ with

$$b_u = (b_u^F; b_u^{\text{cross}}), \quad \text{for all } u \in V.$$

The embeddings $a, b : V \mapsto \mathbb{R}^{3+k}$ defined above is a reachability embedding in G . Notice that

$$\langle a_u, b_v \rangle = \langle a_u^F, b_v^F \rangle + \langle a_u^{\text{cross}}, b_v^{\text{cross}} \rangle.$$

There are three cases to be considered:

- **Case 1:** $u \rightsquigarrow_F v$. The base 3-dimensional embedding on the forest certifies reachability: $\langle a_u^F, b_v^F \rangle > 0$. The additional cross-edge coordinates can only increase the inner product since $\langle a_u^{\text{cross}}, b_v^{\text{cross}} \rangle \geq 0$. Hence $\langle a_u, b_v \rangle > 0$. Finally, any path in F is also a path in G , so $u \rightsquigarrow_F v$ implies $u \rightsquigarrow_G v$.
- **Case 2:** $u \not\rightsquigarrow_F v$ but $u \rightsquigarrow_G v$. Fix any directed path P from u to v in G . Then the path P contains a cross-edge $e_i = (x_i, y_i)$ such that $u \rightsquigarrow_G x_i$ and $y_i \rightsquigarrow_G v$; equivalently, $u \in \text{In}(x_i)$ and $v \in \text{Out}(y_i)$ ¹. By the definition of the cross-edge coordinates, this implies $a_u^{\text{cross}}[i] = b_v^{\text{cross}}[i] = M$, and hence

$$\langle a_u^{\text{cross}}, b_v^{\text{cross}} \rangle \geq a_u^{\text{cross}}[i] \cdot b_v^{\text{cross}}[i] = M^2.$$

Since $\langle a_u^F, b_v^F \rangle \geq \min_{p, q \in V} \langle a_p^F, b_q^F \rangle$, our choice of M ensures

$$\langle a_u, b_v \rangle = \langle a_u^F, b_v^F \rangle + \langle a_u^{\text{cross}}, b_v^{\text{cross}} \rangle > 0.$$

1. Let $P = (u = v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_\ell = v)$ be a directed path in G . If $u \not\rightsquigarrow_F v$, let j be the smallest index such that $u \not\rightsquigarrow_F v_j$. Then $u \rightsquigarrow_F v_{j-1}$ but $u \not\rightsquigarrow_F v_j$. So the edge (v_{j-1}, v_j) is a cross-edge by definition, and its head is v_j .

- **Case 3:** $u \not\rightsquigarrow_G v$. Then in particular $u \not\rightsquigarrow_F v$, and the forest embedding gives $\langle a_u^F, b_v^F \rangle \leq 0$ by Theorem 9. It remains to argue that the cross-edge coordinates do not create a false positive. We claim $\langle a_u^{\text{cross}}, b_v^{\text{cross}} \rangle = 0$: indeed, if there is an index i with $a_u^{\text{cross}}[i] = b_v^{\text{cross}}[i] = M$, then $u \in \text{In}(x_i)$ and $v \in \text{Out}(y_i)$, so $u \rightsquigarrow_G x_i$, $(x_i, y_i) \in E$, and $y_i \rightsquigarrow_G v$, which would imply $u \rightsquigarrow_G v$, a contradiction. Therefore $\langle a_u^{\text{cross}}, b_v^{\text{cross}} \rangle = 0$, and hence

$$\langle a_u, b_v \rangle = \langle a_u^F, b_v^F \rangle + \langle a_u^{\text{cross}}, b_v^{\text{cross}} \rangle \leq 0.$$

■

The following simple observation formalizes why cross-edges are unavoidable on any path that is not realized inside the forest. It is the only place where we use the definition of cross-edges.

Appendix D. Comparing the Treewidth with the Number of Cross-Edges

In this section, we give examples of graph families that have large treewidth but few total number of cross-edges, and vice-versa, showing that these two parameters are incomparable in general.

We first need a basic fact about treewidth.

Lemma 26 (Robertson and Seymour (1986)) *For $n \geq 1$, the complete graph K_n has treewidth $n - 1$.*

We now show our first incomparability result.

Theorem 27 *There exists a DAG G with n nodes such that:*

1. *For some spanning tree T of G , there is no cross-edge w.r.t. T in G .*
2. *The underlying undirected graph G' of G has treewidth $\Omega(n)$.*

Proof We construct $G = (V, E)$ and $T = (V, E_T)$ as follows:

- $V = \{v_1, v_2, \dots, v_n\}$,
- $E = \{(v_i, v_j) \mid 1 \leq i < j \leq n\}$ (i.e. edges point from a smaller label to a larger label),
- $E_T = \{(v_i, v_{i+1}) \mid 1 \leq i \leq n - 1\}$.

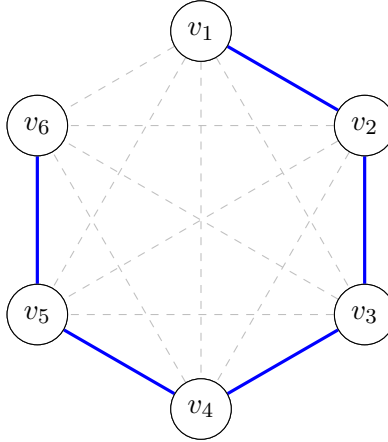
Observe that the vertex v_j is reachable from the vertex v_i in T if and only if $i < j$. Since all directed edges in E are from v_i to v_j for $i < j$, there is no cross-edge w.r.t. T in G . Note that the underlying undirected graph G' is a complete graph K_n . Thus, by Lemma 26, G' has treewidth $\Omega(n)$. See the figure below for an example on 6 nodes where the blue path indicates the spanning tree T and the dashed lines show the cross edges (direction not shown). ■

Now we show the complementary result.

Theorem 28 *For every sufficiently large n , there exists a DAG G with n nodes such that:*

1. *There exists a spanning tree T of G that has $\Omega(n)$ cross-edges w.r.t. T in G .*
2. *The underlying undirected graph G' of G has treewidth $O(1)$.*

Proof We construct such a graph $G = (V, E)$ using a layered structure. Let k be the number of layers, such that the total number of vertices n is linear in k .



Construction Let the layers be L_1, L_2, \dots, L_k .

- $L_1 = \{r\}$ (a single source vertex).
- For $i = 2, \dots, k$, let $L_i = \{u_i, v_i\}$ (two vertices per layer).
- The edge set E consists of all possible edges directed from layer L_i to L_{i+1} . Formally, for each $1 \leq i < k$, there is a directed edge (x, y) for all $x \in L_i$ and $y \in L_{i+1}$.

The total number of vertices is $n = 1 + 2(k - 1)$, so $k = \Theta(n)$.

Lower Bound on Cross-Edges Let T be an arbitrary spanning tree of G rooted at r . Since every vertex $v \in V \setminus \{r\}$ must have exactly one incoming edge in T (its parent), we can count the tree edges layer by layer.

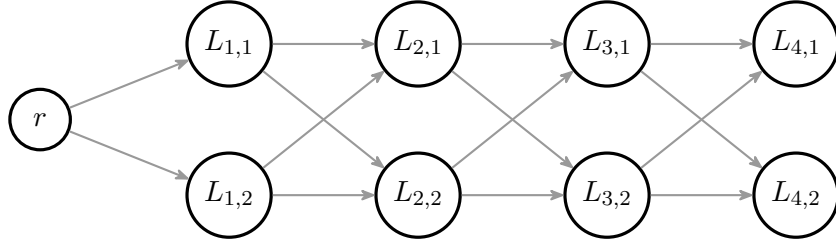
Consider the transition between layer L_i and L_{i+1} for $i \geq 2$. The subgraph induced by $L_i \cup L_{i+1}$ is a complete bipartite graph directed from L_i to L_{i+1} , containing $|L_i| \times |L_{i+1}| = 2 \times 2 = 4$ edges. To include the vertices of L_{i+1} in the spanning tree, T must select exactly 2 edges (one parent for u_{i+1} and one for v_{i+1}). Consequently, 2 edges between these layers are not in T .

In a layered DAG where edges only exist between L_i and L_{i+1} , any edge (x, y) spans exactly one layer. Therefore, x cannot be an ancestor of y unless x is the immediate parent of y . If (x, y) is a non-tree edge, x is not the parent of y , and thus x is not an ancestor of y . Similarly, y cannot be an ancestor of x (due to edge directions). Therefore, all non-tree edges are cross-edges, as desired. Summing over all layers $i = 2$ to $k - 1$, the total number of cross-edges is at least $2(k - 2)$, which is $\Omega(n)$.

Treewidth We show that G' (the underlying undirected graph) has treewidth $O(1)$ by constructing a tree decomposition (specifically, a path decomposition). Define the bags B_1, \dots, B_{k-1} as:

$$B_i = L_i \cup L_{i+1}.$$

It is easy to check that this decomposition satisfies all properties of Definition 3, and the size of each bag is bounded by a constant, implying that the treewidth is $O(1)$. See the figure below for a pictorial representation of the graph. ■



Appendix E. Cross-Edge-Parameterized Lower Bounds

In this section we give two distinct lower bounds for how the required embedding dimension must scale with the number of cross-edges. Recall the definition of cross-edges. Let $F = (V, E_F)$ be a directed spanning-forest subgraph of G , then the cross-edges w.r.t. F are all the edges (u, v) where u cannot reach v in F .

The first lower bound, based on counting the number of matrices of a particular sign rank via Lemma 15, is given below. The first lower bound is useful in the regime when number of cross-edges is large ($\gg n$), and the second one is better suited for small values of the number of cross-edges.

Theorem 29 *Let n be an even number. For every $k \in \{0, 1, \dots, \frac{n^2-2n}{4}\}$, there exists a family $\mathcal{F}_{n,k}$ of DAGs on n vertices such that: (i) each $G \in \mathcal{F}_{n,k}$ satisfies $k(G) = k$, and (ii) $\mathcal{F}_{n,k}$ contains at least $\binom{\Theta(n^2)}{k}$ distinct reachability sign matrices. Consequently, any reachability embedding scheme that succeeds on all graphs in $\mathcal{F}_{n,k}$ must use dimension*

$$d = \Omega\left(\frac{k \log(n^2/k)}{n \log n}\right).$$

Proof Let $m := n/2$ and partition the vertex set into $L = \{\ell_1, \dots, \ell_m\}$ and $R = \{r_1, \dots, r_m\}$. Define the directed edge set

$$E_F := \{(\ell_i, r_i) : i \in [m]\},$$

and let $U := (L \times R) \setminus E_F$, so $|U| = \frac{n^2-2n}{4}$. For each subset $S \subseteq U$ with $|S| = k$, define the bipartite DAG

$$G_S := (L \cup R, E_S), \quad E_S := E_F \cup S.$$

Since all edges go from L to R , every vertex in R has out-degree 0. Hence every directed path has length at most 1, so reachability coincides with adjacency: for $u \in L$ and $v \in R$,

$$u \rightsquigarrow_{G_S} v \iff (u, v) \in E_S.$$

Therefore, distinct choices of S yield distinct reachability relations, and hence distinct reachability sign matrices.

Since $F = (V, E_F)$ is a directed spanning-forest subgraph of G_S , every edge in $E_S \setminus E_F$ is a cross-edge:

$$E_{\text{cross}}(G_S) = E_S \setminus E_F = S, \quad \text{and thus} \quad k(G_S) = |S| = k.$$

It follows that $\mathcal{F}_{n,k} := \{G_S : S \subseteq U, |S| = k\}$ has size $\binom{\Theta(n^2)}{k}$, every graph in the family has k cross-edges and induces distinct reachability sign matrix. Now suppose there is a reachability

embedding of dimension d for every graph in $\mathcal{F}_{n,k}$. Then all $\binom{\Theta(n^2)}{k}$ corresponding reachability sign matrices have sign rank at most d . By Lemma 15, the number of $n \times n$ sign matrices of sign rank at most d is at most $2^{O(dn \log \frac{n}{d})}$. Therefore,

$$\binom{\Theta(n^2)}{k} \leq 2^{O(dn \log \frac{n}{d})},$$

and rearranging and using the fact that $\binom{\Theta(n^2)}{k} \geq \left(\frac{\Theta(n^2)}{k}\right)^k$ gives

$$d = \Omega\left(\frac{k \log(n^2/k)}{n \log n}\right).$$

■

For our second lower bound, we deploy a different strategy. Instead of employing a counting argument to count the number of matrices of a particular sign rank (via Lemma 15), we directly construct a family of graphs with large sign rank. In particular, we construct a DAG whose reachability sign matrix M includes the Sylvester–Hadamard matrix H_t as its submatrix. The Sylvester–Hadamard matrix H_t has spectral norm exactly \sqrt{t} , which allows us to lower-bound the sign-matrix of H_t by utilizing Forster’s Theorem (Theorem 31). Note that the sign-rank of a matrix is at least the sign-rank of any of its submatrices. Thus, we show a lower bound of the sign-rank of M , which immediately yields a lower bound on the embedding dimension d (see the discussion in Section 5). Our second lower bound result of the section is the following.

Theorem 30 *For every sufficiently large k , there exists a DAG with k cross-edges and any reachability embedding for this graph requires dimension $d = \Omega(k^{1/4})$.*

For the sake of completeness, we state Forster’s Theorem (shown as Theorem 31) and the properties of Sylvester–Hadamard matrix.

Theorem 31 (Forster (2002)) *For a matrix $M \in \{-1, 1\}^{n \times n}$, $\text{rank}_{\pm} M \geq n/\|M\|$, where $\|M\|$ represents the spectral norm of M .*

Lemma 32 (Horadam (2012)) *For the Sylvester–Hadamard matrix H_t of order t , where $t = 2^k$ for some positive integer k , we have the following properties:*

1. *The first row and first column of H_t consist entirely of +1 entries.*
2. *Each row (except the first) and each column (except the first) has exactly $t/2$ entries equal to +1 and $t/2$ entries equal to -1.*
3. *The spectral norm of H_t is $\|H_t\| = \sqrt{t}$.*

Furthermore, we need the property that the sign-rank of a matrix is at least the sign-rank of any of its submatrices, the correctness of which can be derived from the definition of sign-rank.

Lemma 33 *Given a matrix $M \in \{-1, 1\}^{n \times n}$. Let M' be a submatrix of M . Then $\text{rank}_{\pm} M \geq \text{rank}_{\pm} M'$.*

Proof Suppose that $\text{rank}_{\pm} M = r$. From the definition of sign-rank, we know that there exists a real matrix R such that $M_{ij} = \text{sign}(R_{ij})$ for all i, j and $\text{rank}(R) = r$. Let R' be the submatrix of R such that $M'_{i'j'} = R'_{i'j'}$ for all i', j' . Since R' is the submatrix of R , then we have $\text{rank}(R') \leq \text{rank}(R) \leq r$. By the definition of sign-rank, we have $\text{rank}_{\pm} M' \leq \text{rank}(R') \leq r = \text{rank}_{\pm} M$. ■

The correctness of Theorem 30 can be directly derived from the following construction.

Lemma 34 *For any $0 < k < n^2/4$, there exists some DAG $G = (V, E)$ with n nodes such that:*

1. *there are less than k cross-edges in G .*
2. *the reachability sign matrix M of G has sign-rank $\Omega(k^{1/4})$.*

Proof Let H_t be a Sylvester-Hadamard matrix of order t . Set $m = \lfloor \frac{\log_2 k}{2} \rfloor$ and $t = 2^m$. Then $t \in (\sqrt{k}/2, \sqrt{k}]$. Since $k < n^2/4$, we have $\sqrt{k} < n/2$, so $n - 2t > 0$.

We construct $G = (V, E)$ as follows:

1. $V = X \cup Y \cup R$, where $X = \{x_i\}_{i=1}^t$, $Y = \{y_j\}_{j=1}^t$ and $R = \{r_s\}_{s=1}^{n-2t}$ are disjoint.
2. $E = \{(x_i, y_j) \mid H_{ij} = 1, \forall i, j \in [t]\}$, where H_{ij} is the (i, j) entry of H_t .

Note that G consists of a bipartite directed graph (X, Y, E) and a set R of isolated vertices.

By the properties of Sylvester-Hadamard matrix shown in Lemma 32, we can know that

1. x_1 can reach every node y_j in Y .
2. For each $i \in [2, t]$, x_i can reach exactly $t/2$ nodes in Y .
3. The spectral norm of H_t is $\|H_t\| = \sqrt{t}$.

Let $F = (V, E')$, where $E' = \{(x_1, y_j) \mid j \in [t]\}$. Then F is a spanning forest of G . Furthermore, there are $(t-1)t/2$ cross-edges. Since $t \leq \sqrt{k}$, the number of cross-edges is $(t-1)t/2 < k$.

Observe that the reachability sign matrix M of G has a sub-matrix H_t , because we can select the rows of x_i and the columns of y_j . By Lemma 33, $\text{rank}_{\pm} M \geq \text{rank}_{\pm} H_t$. By Forster's theorem (refer to Theorem 31), H_t has the sign-rank $\text{rank}_{\pm} H_t \geq t/\|H_t\| = \sqrt{t}$. Therefore, we can conclude that $\text{rank}_{\pm} M \geq \sqrt{t} = \Omega(k^{1/4})$. ■