Uncovering Semantic Hierarchies in Text-Attributed Graphs via Variational EM-based LLM-GNN Synergy

Yunhui Liu[1], **Xudong Jin**[1], **Qizhuo Xie**[1], **Chunhui Zhao**[1]
Xiao Luo[2], **Tao Zheng**[1], **Bin Chong**[3], **Tieke He**[1]*
[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]Department of Statistics, University of Wisconsin–Madison
[3]National Engineering Laboratory for Big Data Analysis and Applications, Peking University
lyhcloudy1225@gmail.com, xiao.luo@wisc.edu, chongbin@pku.edu.cn, hetieke@gmail.com

Abstract

While hierarchical structures are intrinsic to organizing human knowledge, current representation learning on text-attributed graphs predominantly operates on flat semantic spaces, overlooking the rich coarse-to-fine granularity inherent in real-world data. To bridge this gap, we propose SHiFT, which synergizes the reasoning power of LLMs with the structural representation capability of GNNs within an Expectation-Maximization (EM) paradigm. In the E-step, we harness LLMs as expert taxonomists to induce an explicit semantic tree, introducing a lightweight Mapping & Diagnosis strategy that reduces inference costs by surgically updating the hierarchy only when anomalies are detected. In the M-step, we align the GNN encoder with this induced hierarchy from complementary perspectives, including clustering partition, topological skeleton, and semantic concepts. Theoretical analysis interprets SHiFT as a generalized Variational EM algorithm. Extensive experiments demonstrate that SHiFT not only achieves state-of-the-art performance but also uncovers high-quality, human-readable taxonomies. Our code can be found at https://github.com/Cloudy1225/SHiFT.

1 Introduction

Structuring knowledge into hierarchies is fundamental to human cognition [56] and modern AI systems, underpinning biomedical ontologies [60], document taxonomies [10], and recent advances in retrieval-augmented generation [18] and agent memory [49]. Yet hierarchical modeling remains underexplored for Text-Attributed Graphs (TAGs), graphs where nodes carry rich textual content and edges encode semantic relations, which are pervasive in scientific literature, e-commerce, and social media [19]. In these domains, data exhibits natural hierarchical semantics: for instance, in a citation network, *Large Language Models* and *Graph Machine Learning* are distinct sub-fields of *Artificial Intelligence*, residing on different branches from *Computer Vision* [50]. An ideal embedding space should reflect this geometry, encoding node similarity across levels of semantic abstraction.

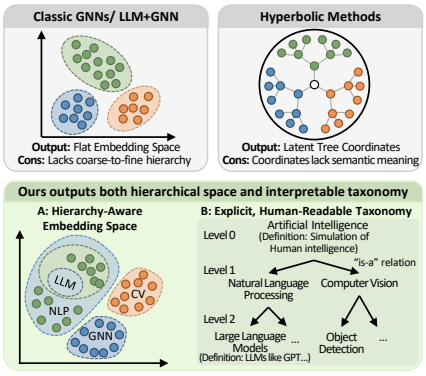


Figure 1: Existing paradigms vs. SHiFT.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

However, prevailing TAG methods, ranging from classic Graph Neural Networks (GNNs) [23] to recent interactions with Large Language Models (LLMs) [19], predominantly operate on flat semantic structures, as shown in Figure 1. While they excel at capturing the graph topology, they often fail to disentangle the semantic hierarchy inherent in the node documents. Existing attempts to introduce hierarchy into graph learning, such as hyperbolic embeddings [2, 68], typically focus on geometric properties. These methods require complex optimization and, crucially, yield latent hierarchies that lack semantic interpretability, e.g., they provide coordinates, not concepts. Consequently, a critical question arises: *How can we effectively uncover and model the explicit semantic hierarchy in TAGs when such taxonomies are absent or incomplete?*

Addressing this question is non-trivial due to the duality of the data: the hierarchy is partially encoded in textual semantics and partially implied by the graph topology, like community structures. Aligning these modalities presents a dilemma. While LLMs possess vast internalized knowledge to recognize semantic hierarchies [21, 73], they lack the ability to perceive the graph structures that reveal subfields. Conversely, GNNs excel at structural clustering [31, 40, 39] but lack the semantic capability to name and organize these clusters into a meaningful taxonomy. Furthermore, naive solutions, such as prompting an LLM to classify every node independently [63], are not only prohibitively expensive but also disregard the relational signals provided by the graph structure.

To bridge this gap, we propose `SHiFT` (`S`emantic `H`ierarchy-`i`nduced `F`ramework for `T`ext-attributed graphs), which synergizes the semantic reasoning of LLMs with the structural representation power of GNNs within an Expectation-Maximization (EM) style framework. In the **E-step**, we harness LLMs as expert taxonomists: hierarchical clustering on the embedding space selects representative coreset documents, from which the LLM induces a tree-structured taxonomy with explicit concept names and definitions; to mitigate inference cost, a lightweight *Mapping & Diagnosis* strategy invokes the LLM only upon detecting anomalies such as concept drift or low cluster cohesion. In the **M-step**, we optimize the GNN encoder via a unified objective that internalizes three facets of the induced taxonomy: *Taxonomy Partition Alignment*, enforcing local cohesion by grouping nodes per cluster assignment; *Taxonomy Skeleton Alignment*, imposing a global geometric constraint so that embedding distances mirror path distances in the hierarchy; and *Taxonomy Concept Alignment*, anchoring node embeddings to LLM-generated concepts across multiple abstraction levels for semantic grounding. This iterative process lets graph topology guide the LLM toward topologically consistent concepts, while the LLM guides the GNN toward a semantically organized space.

We further provide a formal theoretical analysis showing `SHiFT` instantiates a generalized Variational EM algorithm, offering a principled account of its convergence. Experiments show that `SHiFT` not only achieves state-of-the-art performance in classification and clustering tasks but also produces high-quality, human-readable taxonomies that reveal the latent organization of the data.

2 Related Work

2.1 Hierarchical Structure Learning

Capturing hierarchical structures in data has demonstrated broad applicability across domains such as document classification [50, 69], knowledge graph completion [64], retrieval-augmented generation [51, 18], and agent memory systems [49]. One line of research relies on hierarchical contrastive learning [62, 61] or hyperbolic embeddings [4, 15]. However, purely embedding methods lack explicit semantic interpretability (e.g., dimensions in the hyperbolic Poincaré ball do not directly correspond to human-readable concepts). Another line of research involves taxonomy induction [67, 73] and hierarchical clustering [16, 46]. While effective for corpus-level text analysis, these methods typically treat documents as independent instances, ignoring the rich relational signals provided by the graph topology [34]. `SHiFT` bridges these gaps by leveraging graph topology to guide the induction process and utilizing LLMs to generate an explicit, human-readable hierarchy, rather than relying on static external taxonomies or opaque geometric coordinates.

2.2 LLMs for Text-Attributed Graphs

The integration of LLMs into TAG learning has gained significant attention [19, 35]. Existing works mainly fall into three paradigms as follows. (1) *LLM as Encoder*: Using LLMs to generate semantically rich node embeddings, which serve as enhanced input features for downstream GNNs [14, 9, 72].

(2) *LLM as Predictor*: Flattening graph structures into textual prompts, enabling LLMs to directly perform classification [54], zero-shot prediction [71], or explanation generation [45]. (3) *LLM-GNN Alignment*: Employing techniques like contrastive learning to align the latent spaces of GNNs and LLMs, facilitating mutual enhancement [25, 8]. However, most methods operate on a flat semantic plane, neglecting the semantic hierarchy inherent in TAGs [38]. In contrast, `SHiFT` leverages LLMs not as flat predictors, but as efficient taxonomists to induce and refine a global hierarchical taxonomy, which in turn serves as a structural prior to regularize the representation learning process.

3 Preliminaries

In this section, we formally define the problem and provide theoretical motivation for hierarchy-aware embeddings.

3.1 Problem Definition

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{S})$ be a TAG with n nodes, edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, adjacency matrix $\boldsymbol{A} \in \{0,1\}^{n \times n}$. Each node v_i is associated with a raw textual document $s_i \in \mathcal{S}$. Following LLMNodeBed [63], we utilize a frozen pretrained language model (PLM), such as RoBERTa [28], to encode the raw texts into initial node features $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times d_{in}}$. A semantic hierarchy is a rooted tree $\mathcal{T} = (\mathcal{C}, \mathcal{R})$, where each concept $c \in \mathcal{C}$ is characterized by a (name, definition) tuple, and parent-child edges $\mathcal{R}$ encode "is-a" relations. Each graph node $v_i \in \mathcal{V}$ is assigned a unique root-to-leaf path $P(v_i) = \{c_{(0)}, \ldots, c_{(L)}\}$ representing its semantic lineage from the global domain to its finest-grained concept. Given $\mathcal{G}$, our goal is to jointly learn: (1) a GNN encoder $f_\theta : (\mathcal{G}, \boldsymbol{X}) \to \boldsymbol{Z} \in \mathbb{R}^{n \times d}$ mapping nodes to a hierarchy-aware embedding space; and (2) a latent taxonomy $\mathcal{T}$ with assignment $\psi : \mathcal{V} \to \mathrm{Leaves}(\mathcal{T})$ organizing nodes into semantically coherent multi-level clusters.

3.2 Theoretical Motivation: Hierarchy-Aware vs. Flat Embeddings

Real-world TAGs often exhibit intrinsic hierarchical organization, such as paper topics in citation networks, item categories in e-commerce graphs, and user interests in social networks. We formalize this intuition and prove that hierarchy-aware embeddings are provably superior to flat embeddings:

Setup. Let $\mathcal{X}$ be a set of data points whose latent semantic structure is governed by a rooted tree $\mathcal{T}^*$ with K leaf concepts $\{c_1, \ldots, c_K\}$ and depth L. For a point x, let $\ell(x) \in \{1, \ldots, K\}$ denote its ground-truth leaf concept. For two independently sampled points (x_i, x_j), define their latent semantic distance as $D^*_{ij} = d_{\mathcal{T}^*}(\ell(x_i), \ell(x_j))$, where $d_{\mathcal{T}^*}(\cdot,\cdot)$ is the tree-path distance between leaves. For an encoder $f : \mathcal{X} \to \mathbb{R}^d$, we measure how much hierarchy it preserves by the mutual information $\mathcal{I}_{\mathrm{hier}}(f) = I\big(D^*_{ij}; \|f(x_i) - f(x_j)\|_2 \mid \mathcal{T}^*\big)$. This quantity captures whether embedding distances reveal sibling, cousin, or more distant semantic relations in the latent hierarchy.

Theorem 1 (Hierarchy Preservation and Flat Non-identifiability)**.** *Under the hierarchical data-generating process in Definition 1, if a hierarchy-aware encoder f^{hier} is aligned with $\mathcal{T}^*$ with bounded error (ε, r, ζ), sufficient margin Δ and $\varepsilon + 2r < \Delta/2$, then*

$$\mathcal{I}_{\mathrm{hier}}(f^{\mathrm{hier}}) \geq H(D^*_{ij} \mid \mathcal{T}^*) - h_2(2\zeta) - 2\zeta \log(|\mathcal{D}| - 1), \tag{1}$$

where $\mathcal{D}$ is the set of possible tree distances and $h_2(\cdot)$ is the binary entropy function. In contrast, there exists a flat embedding f^{flat} that perfectly separates all leaf concepts but fails to encode hierarchical relations:

$$I\big(D^*_{ij}; \|f^{\mathrm{flat}}(x_i) - f^{\mathrm{flat}}(x_j)\|_2 \mid D^*_{ij} > 0, \mathcal{T}^*\big) = 0. \tag{2}$$

The proof is provided in Appendix A. Theorem 1 shows that flat embeddings may be sufficient for separating leaf clusters but are not identifiable with respect to the semantic skeleton: sibling and distant concepts can collapse to the same inter-cluster distance. By contrast, hierarchy-aware embeddings preserve pairwise tree-distance information, making the learned space more suitable for semantic-rich TAG tasks that depend on the latent concept structure. Therefore, modeling the latent semantic hierarchy is not merely a heuristic design choice but a theoretically motivated idea.

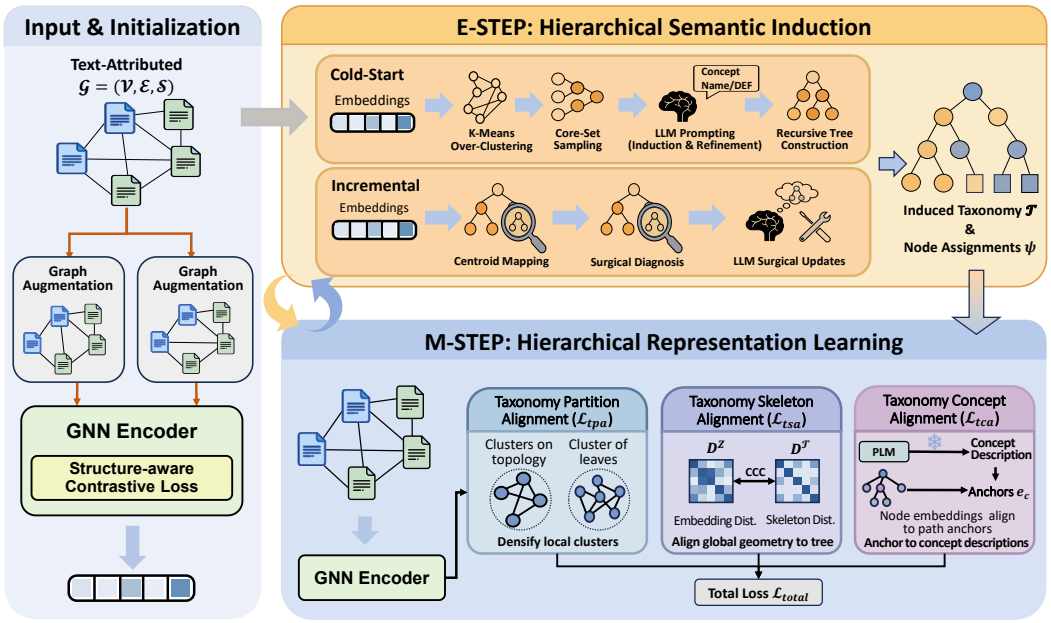


Figure 2: The overview of SHiFT. (Left) The model is initialized via structure-aware pretraining. (Top, **E-Step**) The LLM acts as an expert taxonomist to induce or incrementally update the semantic hierarchy $\mathcal{T}$ based on current node embeddings. (Bottom, **M-Step**) The GNN encoder is optimized to internalize the induced hierarchy through three complementary objectives: Taxonomy Partition Enhancement, Taxonomy Skeleton Alignment, and Taxonomy Concept Alignment.

4 Methodology

In this section, we present SHiFT (Figure 2), which jointly learns a hierarchy-aware graph encoder and an explicit semantic taxonomy within an Expectation-Maximization (EM) paradigm.

4.1 Graph Contrastive Pretraining

Since raw PLM features $\boldsymbol{X}$ lack structural awareness, we perform structure-aware pretraining to initialize the node embeddings [33, 36]. We follow standard graph contrastive learning pipelines [30, 32]: given the graph $\mathcal{G}$, we first generate two augmented views $\mathcal{G}^1, \mathcal{G}^2$ via random edge dropping and feature masking. The GNN encoder f_θ maps these views to ℓ_2-normalized embeddings $\boldsymbol{Z}_1, \boldsymbol{Z}_2 \in \mathbb{R}^{n \times d}$. We minimize a structure-aware contrastive loss:

$$\mathcal{L}_{gcp} = -\sum_{(i,j)\in\mathcal{P}} \boldsymbol{z}_1^i \cdot \boldsymbol{z}_2^j + \gamma \sum_{(i,j)\notin\mathcal{P}} (\boldsymbol{z}_1^i \cdot \boldsymbol{z}_2^j)^2, \tag{3}$$

where $\mathcal{P} = \{(i,j) \mid \boldsymbol{A}_{ij} = 1\}$ denotes the set of positive pairs (including self-loops and connected neighbors) derived from the homophily assumption. This yields structure-enhanced initial embeddings $\boldsymbol{Z}^{(0)}$ for the subsequent E-step.

4.2 E-Step: Hierarchical Semantic Induction

Given the current node embeddings, the E-step estimates a latent semantic hierarchy $\mathcal{T}$ and node-to-leaf assignments ψ. Our goal is not only to partition nodes, but also to induce a human-readable taxonomy. We use embeddings to first cluster the graph into compact semantic groups, and then invoke the LLM on representative documents to construct or update the hierarchy.

4.2.1 Cold-Start Taxonomy Initialization

We initialize the taxonomy in a top-down recursive manner. Starting from the root concept, which represents the global domain, suppose a parent concept c_p contains a document set D_p. We first apply K-Means to the embeddings of nodes in D_p with an intentionally large cluster number, e.g.,

$K_{init} = \sqrt{|D_p|/50}$. This over-clustering step favors high-purity micro-clusters, providing the LLM with localized and coherent evidence rather than a noisy global document collection. Since directly feeding all documents in each cluster to an LLM is infeasible, we then select a small core set $\mathcal{S}_k$ of size M from each micro-cluster C_k. The sampling score combines semantic centrality and structural importance $\text{Score}(v) = a \cdot \cos(\boldsymbol{z}_v, \boldsymbol{\mu}_k) + (1-a) \cdot \text{CCPR}(v \mid C_k)$, where $\boldsymbol{\mu}_k$ is the centroid of C_k, and $\text{CCPR}(v \mid C_k)$ is Cluster-Conditional PageRank computed on the subgraph induced by C_k. Thus, the selected documents are both semantically representative and structurally central.

Given the parent concept c_p and each core set $\mathcal{S}_k$, the LLM acts as a domain taxonomist and generates a candidate child concept with a name, a definition, and a `split_needed` flag indicating whether further refinement is necessary. Because over-clustering may produce redundant or fragmented candidates, we further use an LLM-based editor to refine all candidates under the same parent through three operations: `MERGE` synonymous or overly fine-grained concepts, `PRUNE` outliers irrelevant to the parent domain, and `KEEP` distinct meaningful concepts. The resulting children form a semantically coherent and human-readable expansion of c_p. We recursively apply this procedure to each retained child concept until reaching the maximum depth L_{max}, the minimum cluster size N_{min}, or a negative `split_needed` decision. This yields the initial taxonomy $\mathcal{T}^{(0)}$ and assignment function $\psi^{(0)}$, where each node is associated with a unique root-to-leaf semantic path. Prompts are provided in Appendix D.

4.2.2 Incremental Taxonomy Updates

In subsequent iterations, re-generating the whole taxonomy is unnecessary and costly, since most semantic regions remain stable across iterations. We instead propose a lightweight Mapping & Diagnosis strategy that reuses the previous hierarchy $\mathcal{T}^{(t-1)}$ as a set of semantic anchors, first mapping nodes to existing concepts using the updated embeddings, and then invoking the LLM only when local evidence suggests that the current hierarchy no longer explains the data well.

Specifically, we update the centroid $\boldsymbol{\mu}_c$ of each concept $c \in \mathcal{T}^{(t-1)}$ using the embeddings of its currently assigned nodes. Each node is then reassigned by a top-down greedy search: starting from the root, it is mapped to the child concept whose centroid has the highest cosine similarity, and this process continues until a leaf is reached. If the maximum similarity at any level falls below a threshold δ (e.g., 0.5), the node is placed into an Unassigned Bucket. This centroid-based mapping produces a provisional assignment without requiring LLM calls.

After mapping, we diagnose the taxonomy for four types of semantic inconsistency. First, a *Split* is triggered when a concept exhibits low cohesion, such as a sharp increase in intra-cluster variance or a substantial growth in cluster size; in this case, we rerun the recursive construction procedure within the affected concept to discover finer sub-concepts. Second, a *Merge* is considered when two sibling concepts become highly redundant, e.g., when the cosine similarity between their centroids exceeds τ_{merge}; the LLM is asked to verify whether they denote the same or strongly overlapping concept. Third, a *Redefine* operation is invoked when a concept undergoes significant composition change, such as more than 50% member turnover; the LLM receives the newly representative documents and revises the concept name or definition. Finally, for nodes in the Unassigned Bucket, a *Reassign* operation asks the LLM to either attach them to the most appropriate existing concept or create a new concept if they form a coherent emerging group.

This incremental procedure updates only the semantically unstable parts of the hierarchy while keeping the rest unchanged. As a result, the E-step maintains an explicit and human-readable taxonomy that evolves with the GNN embeddings, but avoids repeatedly querying the LLM over the full graph. The updated hierarchy $\mathcal{T}^{(t)}$ and assignments $\psi^{(t)}$ are then passed to the M-step for hierarchy-aware representation learning.

By limiting LLM calls to these targeted scenarios, Mapping & Diagnosis reduces LLM invocations by $\approx 80\%$ in later epochs while keeping the taxonomy well-aligned with the evolving embedding space (see Appendix G). Full prompts for taxonomy induction are provided in Appendix D. We further present visualizations of induced taxonomies in Appendix K.

4.3 M-Step: Hierarchical Representation Learning

In this step, we fix the taxonomy $\mathcal{T}$ and node assignments ψ derived from the E-step to optimize the GNN encoder f_θ. The core objective is to align the continuous node embeddings with the discrete

hierarchical structure induced by the LLM. We decompose the taxonomy into three distinct facets: the clustering partition, the topological skeleton, and the semantic concepts. Correspondingly, we propose three alignment objectives to internalize these facets into the embedding space:

4.3.1 Taxonomy Partition Alignment (TPA)

The E-step partitions the graph nodes into disjoint semantic groups based on their assignment to leaf concepts. Our first objective serves as a *Local Consistency* constraint: the continuous embedding space must respect these discrete partitions. Unlike prior graph pretraining which relies solely on edge connectivity, TPA explicitly pulls together topologically distant but semantically similar nodes.

We construct a partition-aware adjacency matrix $\boldsymbol{A}^+$, where $\boldsymbol{A}^+_{ij} = 1$ if nodes i and j belong to the same leaf cluster (i.e., $\psi(v_i) = \psi(v_j)$) or are topologically connected. We then employ a contrastive objective to pull intra-partition nodes together while pushing inter-partition nodes apart:

$$\mathcal{L}_{tpa} = -\sum_{(i,j)\in\mathcal{P}^+} \boldsymbol{z}_1^i \cdot \boldsymbol{z}_2^j + \gamma \sum_{(i,j)\notin\mathcal{P}^+} (\boldsymbol{z}_1^i \cdot \boldsymbol{z}_2^j)^2, \tag{4}$$

where $\mathcal{P}^+ = \{(i,j) \mid \boldsymbol{A}^+_{ij} = 1\}$. By optimizing $\mathcal{L}_{tpa}$, the encoder respects the decision boundaries established by the taxonomy's finest granularity.

4.3.2 Taxonomy Skeleton Alignment (TSA)

While TPA ensures local cluster cohesion, it offers no guidance on how these clusters should be arranged relative to one another. To address this, TSA imposes a *Global Geometric* constraint based on the skeleton of the hierarchy tree $\mathcal{T}$. Intuitively, the distances between cluster centroids in the vector space should preserve the path distances in the tree (e.g., sibling concepts should be geometrically closer than cousin concepts).

We enforce this using the Cophenetic Correlation Coefficient (CCC) [52] as a differentiable objective. Let $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$ be the centroid embeddings of the K leaf concepts, calculated as the mean of member node embeddings. We align two distance matrices: (1) The *Skeleton Distance* $\boldsymbol{D}^{\mathcal{T}}$, defined as the shortest path length between concept nodes in $\mathcal{T}$; and (2) The *Embedding Distance* $\boldsymbol{D}^{\boldsymbol{Z}}$, defined as the Euclidean distance $\|\boldsymbol{\mu}_u - \boldsymbol{\mu}_v\|_2$. The alignment loss $\mathcal{L}_{tsa}$ maximizes the Pearson correlation between the upper triangular values of these matrices (i.e., CCC):

$$\mathcal{L}_{tsa} = -\frac{\text{Cov}(\text{vec}(\boldsymbol{D}^{\boldsymbol{Z}}), \text{vec}(\boldsymbol{D}^{\mathcal{T}}))}{\sigma_{\boldsymbol{D}^{\boldsymbol{Z}}} \cdot \sigma_{\boldsymbol{D}^{\mathcal{T}}}}, \tag{5}$$

where $\text{Cov}(\cdot)$ is the covariance, $\text{vec}(\cdot)$ flattens the upper triangle, and σ is the standard deviation. This objective acts as a global regularizer, ensuring the graph embeddings explicitly encode the hierarchical relations (e.g., depth and proximity) inherent in the taxonomy skeleton.

4.3.3 Taxonomy Concept Alignment (TCA)

The previous two objectives focus on the structural alignment (partition and skeleton) but remain agnostic to the concept content. To inject the concept semantics, TCA introduces a *Semantic Grounding* constraint. It ensures that the node embeddings are not just distinct vectors, but are interpretable representations aligned with the LLM-generated definitions.

For each concept $c \in \mathcal{C}$, we encode its text description (name + definition) into a semantic anchor vector $\boldsymbol{e}_c$ using the frozen PLM. For a node v_i, let $P(v_i) = \{c_{(0)}, \ldots, c_{(L)}\}$ be the path from the root to its assigned leaf. We introduce a projection head g_ϕ to map node embeddings to the semantic space and minimize the distance to their conceptual anchors along the hierarchical path:

$$\mathcal{L}_{tca} = \sum_{i\in\mathcal{V}} \sum_{l=1}^{L} \lambda^l \cdot \|g_\phi(\boldsymbol{z}_i) - \boldsymbol{e}_{c_{(l)}}\|_2^2. \tag{6}$$

Here, we employ an exponential growth weight $w_l = \lambda^l$ (with $\lambda > 1$). This design assigns higher importance to finer-grained concepts ($c_{(L)}$) which possess stronger discriminative power, while maintaining a loose constraint on broader categories ($c_{(1)}$), thereby preserving the coarse-to-fine semantic structure.

4.3.4 Optimization and Iteration

The overall M-step objective aggregates these three complementary views:

$$\mathcal{L}_{total} = \mathcal{L}_{tpa} + \alpha\mathcal{L}_{tsa} + \beta\mathcal{L}_{tca}. \tag{7}$$

where α and β are balancing hyperparameters. Through this unified optimization, SHiFT ensures that the learned embeddings are locally partitioned, globally structured by the taxonomy skeleton, and semantically grounded in explicit concepts. Upon convergence of the M-step, the updated embeddings $\boldsymbol{Z}$ are fed back into the E-step, where the Mapping & Diagnosis strategy (as described in Sec 4.2.2) adapts the hierarchy to the evolved embedding space.

Efficiency Analysis. In Appendix G, we analyze the efficiency of SHiFT by presenting its theoretical complexity and empirical runtime comparison against baselines.

4.4 Theoretical Analysis

We further establish a principled probabilistic foundation for SHiFT. Let $\boldsymbol{H}=(\mathcal{T},\psi)$ denote the latent hierarchy and $\boldsymbol{Z}$ the continuous node representations (please kindly see Appendix B for details). We posit that the observed adjacency $\boldsymbol{A}$ is generated via: $p_\theta(\boldsymbol{A})=\sum_{\psi,\mathcal{T}}\int p(\boldsymbol{A}|\boldsymbol{Z},\psi)\,p_\theta(\boldsymbol{Z}|\psi,\mathcal{T})\,p(\psi|\mathcal{T})\,p(\mathcal{T})\,\mathrm{d}\boldsymbol{Z}$, where $p(\mathcal{T})$ encodes the semantic prior, and edges arise from both homophily and partition structure [1]. Under a mean-field approximation $q(\boldsymbol{Z},\psi,\mathcal{T}) = q_\theta(\boldsymbol{Z})q(\psi,\mathcal{T})$ [20] with a Hard-EM approximation [43], the ELBO is given by:

$$\mathcal{L}_{\mathrm{ELBO}} = \underbrace{\mathbb{E}_q[\log p(\boldsymbol{A}|\boldsymbol{Z},\psi)]}_{\text{(I) reconstruction}} + \underbrace{\mathbb{E}_q[\log p_\theta(\boldsymbol{Z}|\psi,\mathcal{T})]}_{\text{(II) embedding prior}} - \underbrace{\mathrm{KL}(q(\psi,\mathcal{T})\,\|\,p(\psi,\mathcal{T}))}_{\text{(III) hierarchy regularization}} + \underbrace{H(q_\theta(\boldsymbol{Z}))}_{\text{(IV) embed. entropy}} . \tag{8}$$

Our two steps perform alternating coordinate ascent on $\mathcal{L}_{\mathrm{ELBO}}$: the **E-step** minimizes Term (III) by searching for $(\hat{\mathcal{T}},\hat{\psi})$ that maximizes the LLM-proxied semantic prior; the **M-step** maximizes Terms (I)–(II), with a precise correspondence: $\mathcal{L}_{tpa}$ operationalizes Term (I) via partition-augmented contrastive learning; $\mathcal{L}_{tsa}$ maximizes Term (II) under a metric-preserving structural prior $p(\boldsymbol{Z}|\mathcal{T})\propto\exp(-\mathrm{dist}(\boldsymbol{D}^{\boldsymbol{Z}},\boldsymbol{D}^{\mathcal{T}}))$; $\mathcal{L}_{tca}$ maximizes Term (III) under a Gaussian content prior $p(\boldsymbol{z}_i|\hat{\psi}_i)=\mathcal{N}(\boldsymbol{z}_i;\boldsymbol{e}_{\hat{\psi}_i},\sigma^2)$. Our LLM-based discrete updates and non-convex neural objectives optimize surrogate terms associated with the same ELBO, leading empirically to stable convergence. The induced hierarchy and learned embeddings progressively become mutually consistent. Full derivations are in Appendix B.

5 Experiments

5.1 Experimental Setup

Datasets & Baselines. We evaluate SHiFT on 5 datasets from the LLMNodeBed benchmark [63], spanning academic, e-commerce, and web domains, and cover a wide range of scales, from 2.7K to 169K nodes. We compare against a diverse set of baselines, from Classic & Hyperbolic GNNs to Hybrid LLM+GNN Methods. Further details are provided in Appendix C.

Implementation Details. Following the LLMNodeBed [63] benchmark, we employ RoBERTa-Large (355M) [28] as the frozen PLM and GCN [23] as the encoder backbone; For the taxonomy induction, we prompt DeepSeek-V3 [7] via API. After representation learning, we attach a linear head and fine-tune for classification under the standard semi-supervised setting. Accuracy is reported with the mean and standard deviation across 5 random seeds. For node clustering, we apply K-Means to the learned embeddings, setting the cluster number to the ground-truth class count. More details are presented in Appendix C.3. Our code can be found at https://github.com/Cloudy1225/SHiFT.

5.2 Evaluation of Node Representations

Node Classification. As shown in Table 1, SHiFT achieves the highest average accuracy, surpassing the strongest baseline TAPE by about 1% and traditional GCN by over 6%. We make three key observations: (1) *Explicit vs. Implicit Hierarchy.* SHiFT significantly outperforms hyperbolic methods

Table 1: Node classification accuracy (%). The **best** and second-best results are highlighted.

Method	Cora	Citeseer	ArXiv	WikiCS	Computer	Avg.
Classic/Hyperbolic GNNs and PLMs						
GCN_{SE} [23]	82.30 ±0.19	70.55 ±0.32	71.39 ±0.28	79.86 ±0.19	71.44 ±1.19	75.11
$SAGE_{SE}$ [13]	82.27 ±0.37	69.56 ±0.43	71.21 ±0.18	79.67 ±0.25	81.43 ±0.27	76.83
GAT_{SE} [59]	81.30 ±0.67	69.94 ±0.74	71.57 ±0.25	79.99 ±0.65	83.39 ±0.22	77.24
HGCN [2]	80.38 ±0.69	70.83 ±0.62	70.39 ±0.38	79.26 ±0.39	79.44 ±0.49	76.06
HGCL [27]	82.37 ±0.47	72.11 ±0.64	71.40 ±0.68	80.16 ±0.35	80.28 ±0.55	77.26
SenBERT [48]	66.66 ±1.42	60.52 ±1.62	72.66 ±0.24	77.77 ±0.75	70.76 ±0.15	69.67
RoBERTa [28]	72.24 ±1.14	66.68 ±2.03	74.12 ±0.12	76.81 ±1.04	72.31 ±0.37	72.43
Hybrid PLM/LLM + GNN Methods						
GIANT [6]	81.08 ±1.07	66.02 ±1.28	74.29 ±0.10	80.48 ±0.65	86.16 ±0.12	77.61
GLEM [70]	81.30 ±0.88	68.80 ±2.46	73.55 ±0.22	76.43 ±0.55	80.46 ±1.45	76.11
ENGINE [74]	84.22 ±0.46	72.14 ±0.74	74.69 ±0.36	80.94 ±0.19	86.42 ±0.23	79.68
GCN_{LLM} [63]	83.33 ±0.75	71.39 ±0.90	74.39 ±0.31	80.94 ±0.16	88.22 ±0.16	79.65
TAPE [14]	84.04 ±0.24	71.87 ±0.35	74.96 ±0.14	81.94 ±0.16	**89.52 ±0.04**	80.47
GAugLLM [9]	82.84 ±0.99	70.19 ±0.92	73.59 ±0.10	80.32 ±0.32	87.79 ±0.23	78.95
GraphGPT [55]	64.72 ±1.50	64.58 ±1.55	75.15 ±0.14	70.71 ±0.37	80.10 ±0.76	71.05
LLaGA [3]	78.94 ±1.14	62.61 ±3.63	74.49 ±0.23	76.47 ±2.20	87.82 ±0.53	76.07
LLM_{IT} [63]	67.00 ±0.16	54.26 ±0.22	76.08 ±0.00	75.02 ±0.16	66.99 ±2.02	67.87
SHiFT **(Ours)**	**84.83 ±0.58**	**73.96 ±0.92**	**76.56 ±0.13**	**82.40 ±0.48**	89.26 ±0.34	**81.40**

(HGCN, HGCL). While hyperbolic spaces implicitly model hierarchy via geometry, SHiFT explicitly injects semantic structure through the taxonomy, proving that semantic grounding is crucial for TAGs. (2) *Synergy of GNN and LLM.* Compared to methods like RoBERTa and LLM_{IT}, which struggle to capture complex graph topology, SHiFT effectively combines the reasoning power of LLMs with the structural propagation of GNNs. (3) *Benefit of Hierarchical Alignment.* Methods like TAPE and GAugLLM use LLMs solely for feature enhancement. By contrast, SHiFT aligns the embedding space with the latent taxonomy skeleton. The performance gain is particularly notable on larger datasets like ArXiv (+1.6% over TAPE), suggesting that the explicit taxonomic guidance helps the GNN navigate complex semantic landscapes better than flat strategies.

Table 2: Node clustering performance (%).

Method	Cora		Citeseer		WikiCS	
	NMI	ARI	NMI	ARI	NMI	ARI
GNN-based Methods						
DMoN [58]	49.00	43.70	39.10	38.40	23.50	9.00
MAGI [29]	52.90	46.40	41.10	39.60	46.70	38.10
LLM+GNN Methods						
GCLR [57]	50.87	59.20	43.30	42.49	32.51	36.49
MARK [12]	57.80	49.00	43.80	43.20	49.30	46.60
SHiFT **(Ours)**	**62.13**	**60.10**	**45.76**	**48.42**	**51.92**	**46.98**

Node Clustering. We report NMI and ARI scores in Table 2. SHiFT demonstrates superior capability in disentangling semantic structures without supervision. Notably, on Cora, SHiFT improves NMI by 4.33% over the recent SOTA LLM+GNN method MARK. This confirms that enforcing local cluster cohesion and global skeleton alignment effectively pulls semantically synonymous nodes together, even when the graph topology is noisy.

Visual Analysis. To intuitively verify that SHiFT captures the intrinsic hierarchy rather than just flat clusters, we visualize the pairwise cosine similarity matrix of the learned embeddings in Appendix J.

5.3 Evaluation of Induced Taxonomy

The benchmarks above provide flat labels only, so we assess the induced taxonomy $\mathcal{T}$ via intrinsic coherence and extrinsic utility, and test it against a known hierarchy in Appendix F. Detailed setups are provided in Appendix E.

Concept Coherence (Intrinsic). We quantify the semantic consistency of the induced concepts using Normalized Pointwise Mutual Information (NPMI). Specifically, we extract top-10 distinct keywords for each concept using Class-based TF-IDF and calculate their pairwise co-occurrence in the corpus (see Appendix E.1). A higher score thus reflects high semantic purity. As shown in Table 3, SHiFT ($\mathcal{T}_{\text{final}}$) consistently outperforms the purely text-based baseline ($\mathcal{T}_{\text{w/o } \mathcal{G}}$, induced solely by PLM

embeddings) and the initial taxonomy ($\mathcal{T}_{\text{init}}$). This confirms that (1) integrating graph structure aids in disambiguating concepts, and (2) our EM-style iteration successfully refines the semantic clusters.

Table 3: Concept Coherence (NPMI) of taxonomies.

Taxonomy	Cora	Citeseer	WikiCS
$\mathcal{T}_{\text{w/o } \mathcal{G}}$	0.1640	0.1602	0.2116
$\mathcal{T}_{\text{init}}$	0.1852	0.1642	0.2204
$\mathcal{T}_{\text{final}}$ (Ours)	**0.1958**	**0.1856**	**0.2410**

Table 4: Human evaluation results (1–5 Likert scale, higher is better). Fleiss' $\kappa = 0.61$.

Taxonomy	Readability	Coherence	HierarchyValidity
$\mathcal{T}_{\text{w/o } \mathcal{G}}$	3.72	3.15	2.87
$\mathcal{T}_{\text{init}}$	4.08	3.83	3.65
$\mathcal{T}_{\text{final}}$ (Ours)	**4.35**	**4.22**	**4.10**

Human Evaluation (Intrinsic). To complement NPMI with a judgment-based assessment, we conducted a small-scale human evaluation. Three CS graduate students evaluated 20 randomly sampled concept nodes on Cora, each presented with the concept name, definition, and 5 representative documents. Annotators rated each concept on three criteria using a 1–5 Likert scale: *Readability* (clarity of concept name/definition), *Coherence* (whether assigned documents genuinely belong to the concept), and *Hierarchy Validity* (logical soundness of parent-child relations). Inter-annotator agreement was substantial (Fleiss' $\kappa = 0.61$) [11]. As shown in Table 4, `SHiFT` consistently outperforms both the text-only baseline and cold-start taxonomy across all criteria. Notably, the largest margin appears in *Hierarchy Validity* ($+1.23$ over text-only $\mathcal{T}_{\text{w/o } \mathcal{G}}$), confirming that graph topology is essential for inducing logically sound hierarchical relations, a quality that NPMI alone cannot fully capture. Full evaluation protocols are in Appendix E.2.

Taxonomy as Features (Extrinsic). To assess whether the taxonomy captures high-level semantics invisible to flat embeddings, we perform a feature augmentation experiment. We convert the hierarchical path of each node into a text string $\mathcal{P}$ and append it to the raw document $\mathcal{S}_{\text{raw}}$ as input for MLP and GCN classifiers (see Appendix E.3). Table 5 reveals two insights: (1) *Semantic Injection:* Appending `SHiFT`-induced paths $\mathcal{P}_{\text{final}}$ to an MLP, which ignores graph structure, boosts accuracy by $\approx 6\%$ on Cora. This implies our taxonomy successfully encodes structural information into explicit textual concepts, allowing the MLP to read the graph structure. (2) *Quality vs. Noise:* Using paths from the graph-agnostic baseline ($\mathcal{P}_{\text{w/o } \mathcal{G}}$) sometimes degrades performance, indicating that pure semantic clustering may create noisy, topologically inconsistent concepts. $\mathcal{P}_{\text{final}}$ consistently boosts GCN performance, proving that our hierarchical concepts provide informative signals complementary to the raw features and adjacency matrix of the graph data.

Table 5: Classification accuracy using taxonomy-augmented features. $\mathcal{P}$ is the path description.

Input Features	Cora	Citeseer	WikiCS
Backbone: MLP (Structure-agnostic)			
$\mathcal{S}_{\text{raw}}$	76.05 ±0.41	67.22 ±0.49	75.79 ±0.18
$\mathcal{S}_{\text{raw}} + \mathcal{P}_{\text{w/o } \mathcal{G}}$	72.68 ±0.34	67.51 ±0.58	77.50 ±0.24
$\mathcal{S}_{\text{raw}} + \mathcal{P}_{\text{init}}$	80.59 ±0.18	71.05 ±0.55	78.06 ±0.37
$\mathcal{S}_{\text{raw}} + \mathcal{P}_{\text{final}}$	**81.91 ±0.55**	**72.83 ±0.73**	**78.33 ±0.21**
Backbone: GCN (Structure-aware)			
$\mathcal{S}_{\text{raw}}$	81.70 ±1.01	69.71 ±0.64	79.95 ±0.15
$\mathcal{S}_{\text{raw}} + \mathcal{P}_{\text{w/o } \mathcal{G}}$	82.02 ±0.54	71.54 ±0.50	79.56 ±0.33
$\mathcal{S}_{\text{raw}} + \mathcal{P}_{\text{init}}$	82.95 ±0.52	70.76 ±0.55	80.28 ±0.34
$\mathcal{S}_{\text{raw}} + \mathcal{P}_{\text{final}}$	**83.41 ±0.44**	**72.08 ±0.51**	**80.78 ±0.18**

Qualitative Analysis. While quantitative metrics reflect statistical cohesion, they may not fully capture structural rationality. We *visually* compare the taxonomies induced by the text-only baseline ($\mathcal{T}_{\text{w/o } \mathcal{G}}$) and `SHiFT` ($\mathcal{T}_{\text{init}}$, $\mathcal{T}_{\text{final}}$) in Appendix K. The final taxonomy exhibits a logical coarse-to-fine progression, resolving semantic ambiguities present in the text-only case.

Verifiable Hierarchy. Appendix F evaluates `SHiFT` on **Patent**, a graph of 10,090 patents with examiner-assigned CPC labels at three levels (7/55/162). The gain over TAPE grows from the section level to the subclass level ($+1.3$ to $+2.6$ accuracy), and hierarchical F-score [24] rises by 2.4 points. The induced tree also tracks the official CPC geometry more closely than a text-only taxonomy.

5.4 Ablation Study

We validate `SHiFT`'s core components against four variants (Table 6): w/o EM (using $\mathcal{T}_{\text{init}}$ without iterative refinement), and removing TPA (partition), TSA (skeleton), or TCA (concept) alignment.

Impact of Iterative Refinement. Removing the EM mechanism (*w/o EM*) leads to significant

degradation across all datasets (e.g., -1.27% avg. drop). This quantitative decline directly validates our qualitative findings in Sec 5.3: the structural evolution from the initialized $\mathcal{T}_{\text{init}}$ to the final $\mathcal{T}_{\text{final}}$ is not merely cosmetic. The iterative refinement allows the GNN to learn from a topologically cleaner hierarchy, proving that the coarse-to-fine granularity is essential for classification accuracy.

Table 6: Ablation study on node classification accuracy.

Method	Cora	Citeseer	WikiCS	Avg Drop
SHiFT **(Full)**	**84.83** ±0.58	**73.96** ±0.92	**82.40** ±0.48	-
w/o EM (One-pass)	83.27 ±0.37	72.62 ±0.75	81.48 ±0.54	↓ 1.27
w/o TPA (Partition)	83.46 ±0.35	72.84 ±0.76	81.18 ±0.22	↓ 1.24
w/o TSA (Skeleton)	83.34 ±0.27	72.31 ±0.94	81.38 ±0.23	↓ 1.39
w/o TCA (Concept)	84.14 ±0.60	73.56 ±0.83	82.05 ±0.44	↓ 0.48

Impact of Alignment Objectives. Deconstructing the M-step reveals the necessity of all three constraints. (1) *Local Cohesion (w/o TPA):* Removing partition alignment consistently hurts performance ($\approx -1.2\%$), as the model loses the explicit guidance to maintain sharp decision boundaries at the leaf level. (2) *Global Structure (w/o TSA):* This variant suffers the most on sparse graphs like Citeseer (-1.65%), confirming that global geometric constraints are critical for preserving inter-community distances in citation networks. (3) *Semantic Grounding (w/o TCA):* While the drop is smaller, it remains significant. Anchoring embeddings to LLM-generated definitions provides semantic robustness, preventing the model from overfitting to pure structural signals.

Supplementary Analysis. Due to space constraints, we provide comprehensive details in the Appendices to support the reproducibility and analysis of our framework. First, Appendix A proves the information-theoretic superiority of hierarchy-aware embeddings; Appendix B offers the formal derivation of SHiFT as a generalized Variational EM algorithm. Regarding experimental reproducibility, Appendices C, D, and E detail the dataset statistics, implementation specifics, full lists of LLM prompts, and the rigorous protocols used for taxonomy evaluation. To validate model robustness, Appendices G, H, and I present efficiency profiling on large-scale graphs, sensitivity analyses across different LLM backbones, and hyperparameter studies. Finally, to enhance interpretability, Appendices J and K provide visual investigations of the embedding space and qualitative case studies of the induced taxonomies. Appendix F reports hierarchical classification and taxonomy recovery on a CPC-annotated patent graph. Appendix L presents the limitations and broader impacts of our work.

6 Conclusion

In this work, we introduce SHiFT to uncover and model the semantic hierarchy in Text-Attributed Graphs. By synergizing the semantic reasoning of LLMs with the structural propagation of GNNs within an Expectation-Maximization paradigm, our approach effectively bridges the gap between flat topological embeddings and rich hierarchical concepts. Specifically, it aligns the continuous embedding space with the discrete taxonomy through partition, skeleton, and concept constraints. We further provide a formal theoretical analysis showing SHiFT instantiates a generalized Variational EM algorithm. Empirical results demonstrate that SHiFT not only achieves remarkable performance on downstream tasks but also yields human-readable taxonomies.

Acknowledgments and Disclosure of Funding

This work was supported by the National Science and Technology Major Project (2026ZD1611200), and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (26CXJH0412), for the authors affiliated with Nanjing University.

References

[1] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.

[2] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

[3] Runjin Chen, Tong Zhao, AJAY KUMAR JAISWAL, Neil Shah, and Zhangyang Wang. LLaGA: Large language and graph assistant. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=B48Pzc4oKi.

[4] Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fully hyperbolic neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5672–5686, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.389. URL https://aclanthology.org/2022.acl-long.389/.

[5] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61, 2024.

[6] Eli Chien, Wei-Cheng Chang, Cho-Jui Hsieh, Hsiang-Fu Yu, Jiong Zhang, Olgica Milenkovic, and Inderjit S Dhillon. Node feature extraction by self-supervised multi-scale neighborhood prediction. In *International Conference on Learning Representations*, 2022.

[7] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

[8] Hanqi Duan, Yao Cheng, Jianxiang Yu, Yao Liu, and Xiang Li. Can large language models act as ensembler for multi-GNNs? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28863–28882, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1470. URL https://aclanthology.org/2025.emnlp-main.1470/.

[9] Yi Fang, Dongzhe Fan, Daochen Zha, and Qiaoyu Tan. GAugLLM: Improving graph contrastive learning for text-attributed graphs with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 747–758, 2024.

[10] Samuel Fernando, Mark Hall, Eneko Agirre, Aitor Soroa, Paul Clough, and Mark Stevenson. Comparing taxonomies for organising collections of documents. In *Proceedings of COLING 2012*, pages 879–894, Mumbai, India, December 2012. URL https://aclanthology.org/C12-1054/.

[11] Fleiss and L. Joseph. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[12] Yiwei Fu, Yuxing Zhang, Chunchun Chen, Jianwen Ma, Quan Yuan, Rong-Cheng Tu, Xinli Huang, Wei Ye, Xiao Luo, and Minghua Deng. MARK: Multi-agent collaboration with ranking guidance for text-attributed graph clustering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6057–6072, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.314. URL https://aclanthology.org/2025.findings-acl.314/.

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 2017.

[14] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: LLM-to-LM interpreter for enhanced text-attributed graph representation learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RXFVcynVe1.

[15] Yuan He, Moy Yuan, Jiaoyan Chen, and Ian Horrocks. Language models as hierarchy encoders. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=GJMYvWzjE1.

[16] Chao-Chun Hsu, Erin Bransom, Jenna Sparks, Bailey Kuehl, Chenhao Tan, David Wadden, Lucy Wang, and Aakanksha Naik. CHIME: LLM-assisted hierarchical organization of scientific studies for literature review support. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 118–132, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.8. URL https://aclanthology.org/2024.findings-acl.8/.

[17] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[18] Haoyu Huang, Yongfeng Huang, Yang Junjie, Zhenyu Pan, Yongqiang Chen, Kaili Ma, Hongzhi Chen, and James Cheng. Retrieval-augmented generation with hierarchical knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6044–6060, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.321. URL https://aclanthology.org/2025.findings-emnlp.321/.

[19] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 2024. doi: 10.1109/TKDE.2024.3469578.

[20] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[21] Priyanka Kargupta, Nan Zhang, Yunyi Zhang, Rui Zhang, Prasenjit Mitra, and Jiawei Han. TaxoAdapt: Aligning LLM-based multidimensional taxonomy construction to evolving research corpora. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29834–29850, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1442. URL https://aclanthology.org/2025.acl-long.1442/.

[22] Thomas N. Kipf and Max Welling. Variational graph auto-encoders, 2016. URL https://arxiv.org/abs/1611.07308.

[23] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[24] Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A Fazel Famili. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Conference of the Canadian society for computational studies of intelligence*, pages 395–406. Springer, 2006.

[25] Yichuan Li, Kaize Ding, and Kyumin Lee. GRENADE: Graph-centric language model for self-supervised representation learning on text-attributed graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2745–2757, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.181. URL https://aclanthology.org/2023.findings-emnlp.181/.

[26] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. In *The Twelfth International Conference on Learning Representations*, 2024.

[27] Jiahong Liu, Menglin Yang, Min Zhou, Shanshan Feng, and Philippe Fournier-Viger. Enhancing hyperbolic graph embeddings via contrastive learning, 2022. URL https://arxiv.org/abs/2201.08554.

[28] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[29] Yunfei Liu, Jintang Li, Yuehe Chen, Ruofan Wu, Ericbk Wang, Jing Zhou, Sheng Tian, Shuheng Shen, Xing Fu, Changhua Meng, et al. Revisiting modularity maximization for graph clustering: A contrastive learning perspective. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1968–1979, 2024.

[30] Yunhui Liu, Xinyi Gao, Tieke He, Tao Zheng, Jianhua Zhao, and Hongzhi Yin. Reliable node similarity matrix guided contrastive graph clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[31] Yunhui Liu, Tieke He, Qing Wu, Tao Zheng, and Jianhua Zhao. Scalable and adaptive spectral embedding for attributed graph clustering. In *Proceedings of the 33rd ACM international conference on information and knowledge management*, pages 3912–3916, 2024.

[32] Yunhui Liu, Tieke He, Tao Zheng, and Jianhua Zhao. Negative-free self-supervised gaussian embedding of graphs. *Neural Networks*, 2024.

[33] Yunhui Liu, Huaisong Zhang, Tieke He, Tao Zheng, and Jianhua Zhao. Bootstrap latents of nodes and neighbors for graph self-supervised learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 76–92. Springer, 2024.

[34] Yunhui Liu, Xinyi Gao, Tieke He, Jianhua Zhao, and Hongzhi Yin. Teaching mlps to master heterogeneous graph-structured knowledge for efficient and accurate inference. *IEEE Transactions on Knowledge and Data Engineering*, 37(10):6189–6201, 2025.

[35] Yunhui Liu, Qizhuo Xie, Jinwei Shi, Jiaxu Shen, and Tieke He. Multi-scale heterogeneous text-attributed graph datasets from diverse domains. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 757–760, 2025.

[36] Yunhui Liu, Jiashun Cheng, Yiqing Lin, Qizhuo Xie, Jia Li, Fugee Tsung, Hongzhi Yin, Tao Zheng, Jianhua Zhao, and Tieke He. Towards anomaly-aware pre-training and fine-tuning for graph anomaly detection. In *International Conference on Learning Representations*, volume 2026, pages 136630–136663, 2026.

[37] Yunhui Liu, Xudong Jin, Kang Zhang, Danshuo An, Yu Xing, Te Song, Jia Liu, and Tieke He. Towards one-for-all foundation model for attributed graph clustering. *arXiv preprint arXiv:2610.07778*, 2026.

[38] Yunhui Liu, Yongchao Liu, Yinfeng Chen, Chuntao Hong, Tao Zheng, and Tieke He. Learning hierarchical knowledge in text-rich networks with taxonomy-informed representation learning. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 961–972, 2026.

[39] Yunhui Liu, Yue Liu, Yongchao Liu, Tao Zheng, Stan Z Li, Xinwang Liu, and Tieke He. Beyond the academic monoculture: a unified framework and industrial perspective for attributed graph clustering. *arXiv preprint arXiv:2603.20829*, 2026.

[40] Yunhui Liu, Pengyu Qiu, Yu Xing, Yongchao Liu, Peng Du, Chuntao Hong, Jiajun Zheng, Tao Zheng, and Tieke He. Bridging academia and industry: a comprehensive benchmark for attributed graph clustering. *arXiv preprint arXiv:2602.08519*, 2026.

[41] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[42] Péter Mernyei and Cătălina Cangea. Wiki-cs: A wikipedia-based benchmark for graph neural networks, 2022. URL https://arxiv.org/abs/2007.02901.

[43] Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[44] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1018. URL https://aclanthology.org/D19-1018/.

[45] Bo Pan, Zhen Xiong, Guanchen Wu, Zheng Zhang, Yifei Zhang, Yuntong Hu, and Liang Zhao. GraphNarrator: Generating textual explanations for graph neural networks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23–42, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.2. URL https://aclanthology.org/2025.acl-long.2/.

[46] Anup Pattnaik, Cijo George, Rishabh Kumar Tripathi, Sasanka Vutla, and Jithendra Vepa. Improving hierarchical text clustering with LLM-guided multi-view cluster representation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 719–727. Association for Computational Linguistics, November 2024. doi: 10.18653/v1/2024.emnlp-industry.54. URL https://aclanthology.org/2024.emnlp-industry.54/.

[47] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250/.

[48] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

[49] Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=moXtEmCleY.

[50] Mobashir Sadat and Cornelia Caragea. Hierarchical multi-label classification of scientific documents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8923–8937. Association for Computational Linguistics, December 2022. doi: 10.18653/v1/2022.emnlp-main.610. URL https://aclanthology.org/2022.emnlp-main.610/.

[51] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GN921JHCRw.

[52] Robert R Sokal and F James Rohlf. The comparison of dendrograms by objective methods. *Taxon*, pages 33–40, 1962.

[53] Robert R Sokal and F James Rohlf. The comparison of dendrograms by objective methods. *Taxon*, 11(2):33–40, 1962.

[54] Yuanfu Sun, Zhengnan Ma, Yi Fang, Jing Ma, and Qiaoyu Tan. GraphICL: Unlocking graph learning potential in LLMs through structured prompt design. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2440–2459, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.131. URL https://aclanthology.org/2025.findings-naacl.131/.

[55] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 491–500, 2024.

[56] Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285, 2011. URL https://www.science.org/doi/10.1126/science.1192788.

[57] Puja Trivedi, Nurendra Choudhary, Edward W Huang, Vassilis N. Ioannidis, Karthik Subbian, and Danai Koutra. Large language model guided graph clustering. In *The Third Learning on Graphs Conference*, 2024. URL https://openreview.net/forum?id=CLyhlb5DG5.

[58] Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. Graph clustering with graph neural networks. *Journal of Machine Learning Research*, 24(127):1–21, 2023. URL http://jmlr.org/papers/v24/20-998.html.

[59] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[60] Lucy Lu Wang, Chandra Bhagavatula, Mark Neumann, Kyle Lo, Chris Wilhelm, and Waleed Ammar. Ontology alignment in the biomedical domain using entity definitions and context. In *Proceedings of the BioNLP 2018 workshop*, pages 47–55. Association for Computational Linguistics, 2018. doi: 10.18653/v1/W18-2306. URL https://aclanthology.org/W18-2306/.

[61] Zhenyi Wang, Yapeng Jia, Haiyan Ning, Peng Wang, Dan Wang, and Yitao Cao. HMCL: Task-optimal text representation adaptation through hierarchical contrastive learning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13495–13518, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.727. URL https://aclanthology.org/2025.findings-emnlp.727/.

[62] Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7109–7119, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.491. URL https://aclanthology.org/2022.acl-long.491/.

[63] Xixi Wu, Yifei Shen, Fangzhou Ge, Caihua Shan, Yizhu Jiao, Xiangguo Sun, and Hong Cheng. When do LLMs help with node classification? a comprehensive analysis. In *Forty-second International Conference on Machine Learning*, 2025.

[64] Qizhuo Xie, Yunhui Liu, Yu Xing, Qianzi Hou, Xudong Jin, Tao Zheng, and Tieke He. Gs-quant: Granular semantic and generative structural quantization for knowledge graph completion. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16782–16797, 2026.

[65] Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, et al. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36:17238–17264, 2023.

[66] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[67] Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian Sadler, Michelle Vanni, and Jiawei Han. Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2701–2709, 2018.

[68] Yifei Zhang, Hao Zhu, Menglin Yang, Jiahong Liu, Rex Ying, Irwin King, and Piotr Koniusz. Understanding and mitigating hyperbolic dimensional collapse in graph contrastive learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1984–1995, 2025.

[69] Yunyi Zhang, Ruozhen Yang, Xueqiang Xu, Rui Li, Jinfeng Xiao, Jiaming Shen, and Jiawei Han. Teleclass: Taxonomy enrichment and llm-enhanced hierarchical text classification with minimal supervision. In *Proceedings of the ACM on Web Conference 2025*, pages 2032–2042, 2025.

[70] Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. Learning on large-scale text-attributed graphs via variational inference. In *The Eleventh International Conference on Learning Representations*, 2023.

[71] Yusheng Zhao, Qixin Zhang, Xiao Luo, Weizhi Zhang, Zhiping Xiao, Wei Ju, Philip S. Yu, and Ming Zhang. Dynamic bundling with large language models for zero-shot inference on text-attributed graphs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=1nSynwHvu2.

[72] Chuang Zhou, Zhu Wang, Shengyuan Chen, Jiahe Du, Qiyuan Zheng, Zhaozhuo Xu, and Xiao Huang. Taming language models for text-attributed graph learning with decoupled aggregation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3463–3474. Association for Computational Linguistics, July 2025. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.173. URL https://aclanthology.org/2025.acl-long.173/.

[73] Kun Zhu, Lizi Liao, Yuxuan Gu, Lei Huang, Xiaocheng Feng, and Bing Qin. Context-aware hierarchical taxonomy generation for scientific papers via LLM-guided multi-aspect clustering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15627–15645, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.788. URL https://aclanthology.org/2025.emnlp-main.788/.

[74] Yun Zhu, Yaoke Wang, Haizhou Shi, and Siliang Tang. Efficient tuning and inference for large language models on textual graphs. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 5734–5742, 2024.

Appendix Table of Contents

To facilitate navigation, we organize the supplementary materials as follows:

- **Appendix A: Theoretical Motivation**
  Formal derivation of the information-theoretic superiority of hierarchy-aware embeddings.
- **Appendix B: Theoretical Analysis**
  Formal derivation of SHiFT as a generalized Variational EM algorithm.
- **Appendix C: Detailed Experimental Setup**
  Dataset statistics, baseline descriptions, and implementation details.
- **Appendix D: List of LLM Prompts**
  Full templates for taxonomy induction, refinement, and diagnosis.
- **Appendix E: Taxonomy Evaluation Protocols**
  Methodologies for calculating Concept Coherence (NPMI) and feature augmentation.
- **Appendix F: Evaluation on a Patent Graph with Verifiable Hierarchy**
  Hierarchical classification and CPC-tree recovery on an examiner-labeled patent graph.
- **Appendix G: Efficiency and Scalability Analysis**
  Runtime profiling and token cost analysis on large-scale graphs.
- **Appendix H: Impact of LLM Selection**
  Ablation studies comparing DeepSeek-V3, GPT-4, and LLaMA-3.
- **Appendix I: Hyperparameter Sensitivity**
  Analysis of taxonomy depth limits and loss balancing weights.
- **Appendix J: Visual Analysis of Representations**
  Heatmaps of embedding similarity and geometric alignment.
- **Appendix K: Qualitative Analysis of Taxonomies**
  Case studies comparing PLM-only vs. SHiFT-induced hierarchies.

A Theoretical Motivation: Proofs and Derivations

We provide formal motivation for why hierarchy-aware representation learning is necessary for text-attributed graphs. A subtle but important point is that the semantic hierarchy $\mathcal{T}^*$ is a global latent structure rather than a per-sample random label. Therefore, instead of measuring $I(\boldsymbol{Z}; \mathcal{T}^*)$ directly, we quantify whether the representation preserves the *hierarchical relations induced by* $\mathcal{T}^*$ among samples. This leads to a pairwise formulation based on tree distances between latent leaf concepts.

A.1 Hierarchical Data-Generating Process

Definition 1 (Hierarchical Data-Generating Process)**.** *Let $\mathcal{T}^* = (\mathcal{C}, \mathcal{R})$ be a rooted tree with depth L and leaf set $\mathcal{Y} = \{1, \ldots, K\}$. For a leaf concept $y \in \mathcal{Y}$, let $\mathrm{anc}_\ell(y)$ denote its ancestor at level ℓ, where $\ell = 0$ is the root and $\ell = L$ is the leaf itself. The generative process is:*

1. ***Hierarchy:*** *A latent semantic hierarchy $\mathcal{T}^*$ is drawn from a prior $p(\mathcal{T}^*)$.*
2. ***Concept assignment:*** *For each data point x_i, a leaf concept $Y_i \sim \mathrm{Cat}(\boldsymbol{\pi})$ is sampled, where $\boldsymbol{\pi}$ is a mixing distribution over leaves. Its level-ℓ semantic label is $Y_i^{(\ell)} = \mathrm{anc}_\ell(Y_i)$.*
3. ***Observation generation:*** *The observed text and graph signal are generated from a conditional distribution $p(x_i \mid Y_i, \mathcal{T}^*)$, whose semantics are determined by the full root-to-leaf path $\{Y_i^{(0)}, \ldots, Y_i^{(L)}\}$.*

For two independently sampled data points (x_i, x_j), define their latent tree distance as $D_{ij}^* = d_{\mathcal{T}^*}(Y_i, Y_j)$, where $d_{\mathcal{T}^*}(\cdot, \cdot)$ is the shortest-path distance between two leaf concepts in $\mathcal{T}^*$. Equivalently, D_{ij}^* encodes the depth of the lowest common ancestor between Y_i and Y_j.

The variable D_{ij}^* is the key object of interest: it captures whether two samples are siblings, cousins, or semantically distant concepts. Thus, a hierarchy-aware representation should not merely separate leaf clusters; it should preserve these pairwise semantic relations.

A.2 Hierarchy-Preserving Representations

Let f_θ be a graph encoder and $\boldsymbol{Z} = f_\theta(\mathcal{G}, \boldsymbol{X})$ be the learned node embeddings. For a node whose latent leaf concept is Y, write its embedding as $\boldsymbol{z}$. We measure the amount of hierarchy preserved by f_θ through the mutual information between the latent tree distance and the embedding-pair statistic:

$$\mathcal{I}_{\text{hier}}(f_\theta) = I\big(D_{ij}^*; \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 \mid \mathcal{T}^*\big)\,. \tag{9}$$

This quantity is well-defined because D_{ij}^* is a random variable induced by randomly sampled data points, while $\mathcal{T}^*$ is treated as the conditioning latent hierarchy. A large value of $\mathcal{I}_{\text{hier}}(f_\theta)$ means that distances in the learned embedding space reveal the semantic distance in the underlying hierarchy.

Definition 2 (Hierarchy-aligned prototype representation)**.** *Let $\boldsymbol{\mu}_y \in \mathbb{R}^d$ denote the prototype embedding of leaf concept y. A representation is (ε, r, ζ)-hierarchy-aligned if there exists a strictly increasing function $g : \mathcal{D} \to \mathbb{R}_+$ over the finite set $\mathcal{D} = \{d_{\mathcal{T}^*}(y, y') : y, y' \in \mathcal{Y}\}$ such that:*

1. ***Prototype tree-distance alignment:***

$$\big|\|\boldsymbol{\mu}_y - \boldsymbol{\mu}_{y'}\|_2 - g(d_{\mathcal{T}^*}(y, y'))\big| \le \varepsilon, \quad \forall y, y' \in \mathcal{Y}. \tag{10}$$

2. ***Within-concept concentration:*** *for a sample embedding $\boldsymbol{z}$ with latent leaf $Y = y$,*

$$\Pr\big(\|\boldsymbol{z} - \boldsymbol{\mu}_y\|_2 \le r \mid \mathcal{T}^*\big) \ge 1 - \zeta. \tag{11}$$

Let

$$\Delta_g = \min_{d \ne d' \in \mathcal{D}} |g(d) - g(d')| \tag{12}$$

be the minimum separation between distinct tree-distance levels after transformation by g.

Theorem 2 (Hierarchy alignment preserves tree-distance information)**.** *Suppose a representation is (ε, r, ζ)-hierarchy-aligned and satisfies*

$$\varepsilon + 2r < \frac{\Delta_g}{2}. \tag{13}$$

Then the latent tree distance D_{ij}^* can be recovered from the embedding distance $\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2$ with error probability at most 2ζ. Consequently,

$$\mathcal{I}_{\text{hier}}(f_\theta) \ge H(D_{ij}^* \mid \mathcal{T}^*) - h_2(2\zeta) - 2\zeta \log(|\mathcal{D}| - 1), \tag{14}$$

where $h_2(\cdot)$ is the binary entropy function. In particular, when $\zeta = 0$, the representation preserves the full tree-distance information:

$$\mathcal{I}_{\text{hier}}(f_\theta) = H(D_{ij}^* \mid \mathcal{T}^*). \tag{15}$$

Proof. Consider two samples with latent leaves $Y_i = y$ and $Y_j = y'$. When both embeddings satisfy the concentration event in Eq. (11), the triangle inequality gives

$$\begin{aligned}\Big|\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 - \|\boldsymbol{\mu}_y - \boldsymbol{\mu}_{y'}\|_2\Big| &\le \|\boldsymbol{z}_i - \boldsymbol{\mu}_y\|_2 + \|\boldsymbol{z}_j - \boldsymbol{\mu}_{y'}\|_2 \\ &\le 2r.\end{aligned} \tag{16}$$

Combining this with the prototype alignment condition in Eq. (10), we obtain

$$\Big|\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 - g(D_{ij}^*)\Big| \le \varepsilon + 2r. \tag{17}$$

By the margin condition in Eq. (13), the intervals centered at $\{g(d) : d \in \mathcal{D}\}$ with radius $\varepsilon + 2r$ are disjoint. Therefore, the nearest-distance decoder

$$\widehat{D}_{ij} = \arg\min_{d \in \mathcal{D}} \big|\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 - g(d)\big| \tag{18}$$

recovers D^*_{ij} whenever both concentration events hold. By the union bound, the probability that at least one of the two concentration events fails is at most 2ζ. Thus,

$$\Pr(\widehat{D}_{ij} \neq D^*_{ij}) \leq 2\zeta. \tag{19}$$

Applying Fano's inequality yields

$$H(D^*_{ij} \mid \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2, \mathcal{T}^*) \leq h_2(2\zeta) + 2\zeta \log(|\mathcal{D}| - 1). \tag{20}$$

Finally, by the definition of mutual information,

$$\begin{aligned} I(D^*_{ij}; \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 \mid \mathcal{T}^*) = H(D^*_{ij} \mid \mathcal{T}^*) \\ - H(D^*_{ij} \mid \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2, \mathcal{T}^*), \end{aligned} \tag{21}$$

which gives Eq. (14). When $\zeta = 0$, the recovery error is zero, and the conditional entropy term vanishes. □

A.3 Why Flat Partition Objectives Are Insufficient

Theorem 2 shows that hierarchy-aligned embeddings preserve the latent tree-distance structure. We next show that a flat partition-preserving representation can perfectly separate leaf concepts while still failing to encode their hierarchical relations.

Theorem 3 (Non-identifiability of hierarchy under flat partition alignment)**.** *Assume $K \geq 3$ leaf concepts. There exists a representation that perfectly separates all leaf concepts but contains no information about positive tree distances in its embedding distances. Specifically, there exist prototypes $\{\boldsymbol{\mu}_y\}_{y=1}^K$ such that*

$$\|\boldsymbol{\mu}_y - \boldsymbol{\mu}_{y'}\|_2 = q, \quad \forall y \neq y', \tag{22}$$

for some constant $q > 0$. For embeddings concentrated around these prototypes, the distance statistic $\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2$ can distinguish whether two samples belong to the same leaf, but cannot distinguish different positive tree distances. In the noiseless case,

$$I\big(D^*_{ij}; \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 \mid D^*_{ij} > 0, \mathcal{T}^*\big) = 0. \tag{23}$$

Proof. A regular simplex in $\mathbb{R}^{K-1}$ provides K prototypes $\{\boldsymbol{\mu}_y\}_{y=1}^K$ satisfying Eq. (22). Thus, every pair of distinct leaf concepts has exactly the same prototype distance q. This representation separates leaves perfectly because $\boldsymbol{\mu}_y \neq \boldsymbol{\mu}_{y'}$ for all $y \neq y'$. Therefore, any objective that only enforces leaf-level partition consistency can be minimized.

However, conditioned on $D^*_{ij} > 0$, the two samples come from different leaves. In the noiseless case, their embedding distance is always q, regardless of whether the leaves are siblings, cousins, or semantically distant concepts in $\mathcal{T}^*$. Hence,

$$\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 = q \quad \text{for all } D^*_{ij} > 0. \tag{24}$$

The distance statistic is therefore constant after conditioning on $D^*_{ij} > 0$, and a constant random variable has zero mutual information with D^*_{ij}. This proves Eq. (23). □

Implication for `SHiFT`. Theorem 3 shows that flat partition alignment alone cannot identify the semantic skeleton of the hierarchy: leaf clusters may be perfectly separated while all inter-cluster relations remain geometrically indistinguishable. Theorem 2 further shows that if embedding distances are aligned with tree distances, then the representation preserves the hierarchical relations induced by $\mathcal{T}^*$. This motivates the three complementary objectives in the M-step of `SHiFT`: TPA enforces local partition consistency, TSA aligns global embedding geometry with the taxonomy skeleton, and TCA anchors the learned space to explicit LLM-generated semantic concepts.

B Theoretical Analysis: Proofs and Derivations

In this section, we provide a probabilistic interpretation of `SHiFT`. We demonstrate that our iterative framework constitutes a Variational Expectation-Maximization algorithm, where the proposed objectives maximize the Evidence Lower Bound of the log-likelihood of the graph structure.

B.1 Probabilistic Generative Model

Following the paradigm of Variational Graph Auto-Encoders [22], we posit that the observed graph adjacency $\boldsymbol{A}$ is generated from latent variables. Let $\boldsymbol{H} = (\mathcal{T}, \psi)$ denote the latent hierarchy variables, where $\mathcal{T}$ represents the taxonomy structure (comprising tree skeleton $\mathcal{R}$ and concept definitions $\mathcal{C}$) and $\psi : \mathcal{V} \to \text{Leaves}(\mathcal{T})$ denotes the discrete node-to-concept assignment (partition). Let $\boldsymbol{Z}$ denote the continuous latent node representations. The generative process is defined as follows:

1. **Taxonomy Generation:** A taxonomy $\mathcal{T}$ is drawn from a semantic prior $p(\mathcal{T})$, which governs the valid semantic organization of the domain (simulated by the LLM).
2. **Partition Generation:** For each node v_i, a concept assignment ψ_i is sampled based on the hierarchy $\mathcal{T}$. This determines the semantic identity of the node before its features are instantiated.
3. **Embedding Generation:** The continuous representation $\boldsymbol{z}_i$ is generated conditioned on its assigned concept ψ_i and the taxonomy $\mathcal{T}$, following a conditional likelihood $p_\theta(\boldsymbol{z}_i|\psi_i, \mathcal{T})$. For instance, $\boldsymbol{z}_i$ is drawn from a distribution centered at the semantic anchor of concept ψ_i.
4. **Observation Generation:** Finally, the graph structure $\boldsymbol{A}$ is generated based on node embeddings and their partitions, governed by $p(\boldsymbol{A}|\boldsymbol{Z}, \psi)$ [22]. Consistent with stochastic block models [1] or homophily assumptions [41], edges are more likely to form between nodes in the same semantic partition or with high embedding similarity.

Our goal is to maximize the marginal log-likelihood of the observed graph structure:

$$\log p_\theta(\boldsymbol{A}) = \log \sum_{\psi, \mathcal{T}} \int p(\boldsymbol{A}, \boldsymbol{Z}, \psi, \mathcal{T}) \, \mathrm{d}\boldsymbol{Z}. \tag{25}$$

B.2 Evidence Lower Bound (ELBO)

Direct maximization of the marginal likelihood is intractable due to the integration over continuous embeddings $\boldsymbol{Z}$ and the summation over the combinatorial space of hierarchies. We introduce a variational distribution $q(\boldsymbol{Z}, \psi, \mathcal{T})$ to approximate the true posterior. Following mean-field variational inference [20], we factorize it as $q(\boldsymbol{Z}, \psi, \mathcal{T}) = q_\theta(\boldsymbol{Z})q(\psi, \mathcal{T})$. Here, $q_\theta(\boldsymbol{Z})$ denotes the variational posterior over continuous node representations, whose mean is parameterized by the GNN encoder.

Starting from the marginal likelihood, we have:

$$\begin{aligned} \log p(\boldsymbol{A}) &= \log \sum_{\psi, \mathcal{T}} \int p(\boldsymbol{A}, \boldsymbol{Z}, \psi, \mathcal{T}) \, \mathrm{d}\boldsymbol{Z} \\ &= \log \sum_{\psi, \mathcal{T}} \int q_\theta(\boldsymbol{Z}) q(\psi, \mathcal{T}) \frac{p(\boldsymbol{A}, \boldsymbol{Z}, \psi, \mathcal{T})}{q_\theta(\boldsymbol{Z}) q(\psi, \mathcal{T})} \, \mathrm{d}\boldsymbol{Z}. \end{aligned} \tag{26}$$

By Jensen's inequality, the ELBO is:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_q[\log p(\boldsymbol{A}, \boldsymbol{Z}, \psi, \mathcal{T})] - \mathbb{E}_q[\log q_\theta(\boldsymbol{Z})] - \mathbb{E}_q[\log q(\psi, \mathcal{T})]. \tag{27}$$

Using the factorization

$$p(\boldsymbol{A}, \boldsymbol{Z}, \psi, \mathcal{T}) = p(\boldsymbol{A}|\boldsymbol{Z}, \psi) p(\boldsymbol{Z}|\psi, \mathcal{T}) p(\psi|\mathcal{T}) p(\mathcal{T}), \tag{28}$$

we obtain:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_q[\log p(\boldsymbol{A}|\boldsymbol{Z}, \psi)] + \mathbb{E}_q[\log p(\boldsymbol{Z}|\psi, \mathcal{T})] - \mathrm{KL}\big(q(\psi, \mathcal{T}) \,\|\, p(\psi, \mathcal{T})\big) + H(q_\theta(\boldsymbol{Z})), \tag{29}$$

where $p(\psi, \mathcal{T}) = p(\psi|\mathcal{T})p(\mathcal{T})$ and $H(q_\theta(\boldsymbol{Z})) = -\mathbb{E}_{q_\theta}[\log q_\theta(\boldsymbol{Z})]$ is the entropy of the continuous variational posterior.

In practice, `SHiFT` uses a deterministic GNN encoder. This can be interpreted as using a fixed-variance Gaussian posterior:

$$q_\theta(\boldsymbol{Z}) = \prod_{i=1}^{n} \mathcal{N}\left(\boldsymbol{z}_i; f_\theta(\mathcal{G}, \boldsymbol{X})_i, \sigma_q^2 \boldsymbol{I}\right), \tag{30}$$

and evaluating the expected terms at the posterior mean. Since σ_q^2 is fixed, the entropy term becomes:

$$H(q_\theta(\boldsymbol{Z})) = \frac{nd}{2}\left(1 + \log(2\pi\sigma_q^2)\right), \tag{31}$$

which is constant with respect to the encoder parameters. Therefore, although the entropy term is part of the formal ELBO, it does not affect the practical optimization objective and can be safely absorbed into the constant term.

B.3 E-Step: Taxonomy Induction

In the E-step, we fix the encoder parameters and thus the current variational posterior $q_\theta(\boldsymbol{Z})$. We then optimize the discrete variational distribution $q(\psi, \mathcal{T})$. Following the Hard-EM approximation [43], we collapse $q(\psi, \mathcal{T})$ to a point estimate:

$$q(\psi, \mathcal{T}) = \delta_{(\hat{\psi}, \hat{\mathcal{T}})}(\psi, \mathcal{T}). \tag{32}$$

Under this approximation, the entropy of $q(\psi, \mathcal{T})$ becomes zero, and the hierarchy-related part of the ELBO reduces to finding a taxonomy and node assignment that are both compatible with the current embeddings and likely under the semantic prior:

$$(\hat{\psi}, \hat{\mathcal{T}}) \approx \arg\max_{\psi, \mathcal{T}} \mathbb{E}_{q_\theta(\boldsymbol{Z})}[\log p(\boldsymbol{Z}|\psi, \mathcal{T})] + \log p(\psi, \mathcal{T}). \tag{33}$$

The first term encourages the induced hierarchy to explain the current embedding distribution, while the second term enforces semantic plausibility. Since the semantic prior $p(\mathcal{T})$ is analytically unavailable, `SHiFT` uses the LLM as a proxy for this prior [47]. The Mapping phase finds a provisional assignment based on embedding similarity, while the Diagnosis phase invokes the LLM to refine semantically unstable regions, thereby approximating the discrete E-step.

B.4 M-Step: Representation Learning

In the M-step, we fix the hierarchy $\hat{\mathcal{T}}$ and partition $\hat{\psi}$ obtained from the E-step, and optimize the encoder parameters θ. Since $q_\theta(\boldsymbol{Z})$ is modeled as a fixed-variance Gaussian centered at the deterministic GNN outputs, the entropy term $H(q_\theta(\boldsymbol{Z}))$ is constant. Therefore, optimizing θ amounts to maximizing the following ELBO components:

$$\mathbb{E}_{q_\theta}[\log p(\boldsymbol{A}|\boldsymbol{Z}, \hat{\psi})] + \mathbb{E}_{q_\theta}[\log p(\boldsymbol{Z}|\hat{\psi}, \hat{\mathcal{T}})]. \tag{34}$$

We show that our three alignment objectives naturally correspond to these two components.

Partition Alignment ($\mathcal{L}_{tpa}$) $\Leftrightarrow \log p(\boldsymbol{A}|\boldsymbol{Z}, \hat{\psi})$. This term maximizes the likelihood of the graph structure. We assume a generative mechanism where edges arise from both pairwise embedding similarity, corresponding to homophily [41], and shared semantic partitions, corresponding to community structure [1]. Consequently, a high likelihood $p(\boldsymbol{A}|\boldsymbol{Z}, \hat{\psi})$ implies that nodes should be close in the embedding space if they are either topologically connected or assigned to the same semantic cluster. Our $\mathcal{L}_{tpa}$ operationalizes this by constructing an augmented adjacency $\boldsymbol{A}^+$ that includes both observed edges and intra-partition pairs. By treating these pairs as positives in the contrastive objective, the encoder respects the observed topology while enforcing consistency with the latent partition derived in the E-step.

Embedding Prior Matching $\Leftrightarrow \log p(\boldsymbol{Z}|\hat{\psi}, \hat{\mathcal{T}})$. The second term constrains node embeddings to follow the hierarchical prior implied by the induced taxonomy. We decompose this prior into structural and semantic components:

- **Skeleton Alignment ($\mathcal{L}_{tsa}$):** We model the structural prior as preserving the tree metric:

$$p(\boldsymbol{Z}|\hat{\mathcal{T}}) \propto \exp\left(-\mathrm{dist}(\boldsymbol{D}^{\boldsymbol{Z}}, \boldsymbol{D}^{\hat{\mathcal{T}}})\right). \tag{35}$$

  Maximizing this likelihood corresponds to aligning the embedding geometry with the taxonomy skeleton. $\mathcal{L}_{tsa}$ implements this alignment through the Cophenetic correlation objective.

- **Concept Alignment ($\mathcal{L}_{tca}$):** We model the semantic content prior as Gaussian distributions centered at LLM-generated concept anchors:

$$p(\boldsymbol{z}_i|\hat{\psi}_i) = \mathcal{N}(\boldsymbol{z}_i; \boldsymbol{e}_{\hat{\psi}_i}, \sigma^2 \boldsymbol{I}). \tag{36}$$

  Maximizing the corresponding log-likelihood is equivalent to minimizing the Euclidean distance between a node representation and its assigned concept anchor. $\mathcal{L}_{tca}$ extends this idea to all concepts along the root-to-leaf path, thereby enforcing multi-level semantic grounding.

B.5 Convergence Discussion

The above derivation shows that `SHiFT` can be interpreted as a generalized hard Variational EM algorithm [43]. The E-step updates the discrete hierarchy variables $(\hat{\psi}, \hat{\mathcal{T}})$, while the M-step updates the continuous encoder parameters θ. The entropy term $H(q_\theta(\boldsymbol{Z}))$ is part of the formal ELBO, but becomes constant under our fixed-variance posterior approximation and thus does not affect optimization. Unlike classical EM, however, `SHiFT` does not provide a strict monotonic convergence guarantee. This is because the E-step involves LLM-based discrete taxonomy edits, which approximate rather than exactly solve the posterior update, and the M-step optimizes non-convex neural objectives that serve as tractable surrogates of the likelihood terms. Therefore, `SHiFT` should be viewed as a generalized surrogate EM procedure rather than an exact EM algorithm. Empirically, the alternating updates lead to stable convergence, as the induced hierarchy and learned embeddings progressively become mutually consistent.

C Detailed Experimental Setup

C.1 Datasets

To provide a comprehensive evaluation of our proposed method, we adopt five datasets from the `LLMNodeBed` [63] benchmark, as summarized in Table 7. These datasets span diverse domains, including academic networks, web graphs, and product co-purchase networks, and cover a wide range of scales, from thousands to hundreds of thousands of nodes. Within `LLMNodeBed`, each dataset is stored in `.pt` format using PyTorch, which includes shallow embeddings, raw text of nodes, edge indices, labels, and data splits for convenient loading. The processed data is publicly available at `https://huggingface.co/datasets/xxwu/LLMNodeBed` under the MIT license. A description of these datasets is provided below:

- **Academic Networks:** The **Cora**, **Citeseer** [23], and **ogbn-arXiv** (abbreviated as "ArXiv") [17] datasets consist of nodes representing papers, with edges indicating citation relationships. The associated text attributes include each paper's title and abstract, which were collected by He et al. [14] and Chen et al. [5], respectively. Within the dataset, each node is labeled according to its category. For example, the ArXiv dataset includes 40 CS sub-categories such as cs.AI (Artificial Intelligence) and cs.DB (Databases).
- **Web Link Network:** In the **WikiCS** dataset [42], each node represents a Wikipedia page, and edges indicate reference links between pages. The raw text for each node includes the page name and content, which was collected by Liu et al. [26]. The classification goal is to categorize each entity into different Wikipedia categories.
- **E-Commerce Networks:** The **Ele-Computer** (abbreviated as "Computer") dataset is derived from the Amazon Electronics dataset [44], where each node represents an item in the Computer category. We utilize the processed datasets released in Yan et al. [65]. In this e-commerce network, edges indicate co-purchase or co-view relationships. The associated text for each item includes descriptions or user reviews. The classification task involves categorizing the products into fine-grained sub-categories.

These five datasets release flat class labels only. That does not mean the underlying domains are unstructured: citation networks are organized in practice by taxonomies such as the ACM Computing Classification System, and product graphs by catalogs such as the Amazon product taxonomy, but those hierarchical annotations are not part of the public splits. Flat accuracy and clustering gains are therefore consistent with hierarchy-aware representations capturing latent structure, yet they do not directly test recovery of a known tree. Appendix F supplies that test on a patent graph whose CPC labels are assigned by examiners.

Table 7: Dataset statistics including the number of nodes, edges, and classes, and their domains.

Dataset	#Nodes	#Edges	#Classes	Domain
Cora	2,708	5,429	7	Citation
Citeseer	3,186	4,277	6	Citation
ArXiv	169,343	1,166,243	40	Citation
WikiCS	11,701	216,123	10	Web Link
Computer	87,229	721,081	10	E-Commerce

C.2 Baselines

Following the `LLMNodeBed` [63] benchmark, we compare our method against the following categories of approaches:

- **Classic GNNs with Shallow Embeddings:** GCN_{SE} [23], SAGE_{SE} [13], and GAT_{SE} [59], which use traditional GNN architectures combined with shallow text features such as bag-of-words.
- **Hyperbolic GNNs:** HGCN [2] and HGCL [27], which map data into hyperbolic space to capture hierarchy implicitly.
- **PLM-based Methods:** Fine-tuned encoder-only pre-trained language models used to obtain text embeddings for classification, including SenBERT (66M) [48] and RoBERTa (355M) [28].
- **Hybrid PLM + GNN:** Models integrating PLM-derived textual embeddings with GNN-based structural learning. We evaluate GIANT [6] and GLEM [70] as representative methods.
- **LLM-as-Encoder:** Approaches that use large language models as node feature encoders, injecting LLM-derived embeddings into graph learning pipelines. We include ENGINE [74] and GCN_{LLM} [63].
- **LLM-as-Explainer:** Methods that leverage LLMs to generate auxiliary semantic explanations to enhance node representations. We consider TAPE [14] and GAugLLM [9].
- **LLM-as-Predictor:** Approaches where LLMs directly perform node classification via prompts or instruction-tuned generation. We include GraphGPT [55], LLaGA [3], and instruction-tuned LLMs (LLM_{IT}) [63].

C.3 Implementation Details

C.3.1 Architecture Setup

To ensure a fair comparison with state-of-the-art methods, we align our architectural choices with the `LLMNodeBed` benchmark suite [63].

- **Text Encoder (PLM):** We utilize the frozen **RoBERTa-Large** (355M parameters) [28] to encode raw node documents into initial feature vectors.
- **Graph Encoder (GNN):** We employ a standard **GCN** [23] as the backbone encoder. Following the benchmark, we perform a grid search for the optimal architecture: number of layers $L \in \{2, 3\}$, hidden dimension $d \in \{64, 128, 256\}$, with optional batch normalization and residual connections.
- **LLM for Taxonomy Induction:** For the semantic reasoning in the E-step, we access **DeepSeek-V3** [7] via API. To ensure reproducibility, we set the generation temperature to 0.1.

C.3.2 Algorithm Configuration

The training process of `SHiFT` involves a warm-up phase followed by the iterative EM loop. The specific configurations for each stage are as follows:

Warm-up: Graph Contrastive Pretraining. Before the EM iterations, we initialize the node embeddings via structure-aware contrastive learning (Eq. (3)). We pretrain the GNN encoder for up to 50 epochs using the Adam optimizer. The contrastive weight γ is set to 1.0.

E-Step: Hierarchical Semantic Induction. We construct the taxonomy with a maximum depth of $L_{max} = 3$. To ensure statistical significance of the induced concepts, we enforce the following constraints during the recursive clustering and LLM diagnosis:

- **Minimum Cluster Size:** A valid concept node must contain at least $N_{min} = 50$ document samples.
- **Merge Threshold:** Two sibling concepts are merged if their centroid cosine similarity exceeds $\tau_{merge} = 0.9$, indicating semantic redundancy.
- **Split Criterion:** A cluster is flagged for splitting if its internal variance increases by a factor of $1.5\times$ compared to the previous iteration, suggesting low cohesion.
- **Pruning:** Degenerate clusters with fewer than 5 members are pruned as outliers.

M-Step: Hierarchical Representation Learning. Fixing the induced taxonomy, we optimize the GNN encoder to minimize the joint objective $\mathcal{L}_{total}$. To mitigate the quadratic complexity of contrastive terms, we employ **negative sampling** for both the pretraining (Eq. (3)) and partition alignment (Eq. (4)) objectives. Specifically, for each positive pair, we sample exactly one negative pair (1:1 ratio). We train for 10 epochs per iteration. The balancing hyperparameters are set by default: Taxonomy Partition Alignment weight $\gamma = 1.0$, Taxonomy Skeleton Alignment weight $\alpha = 0.5$, and Taxonomy Concept Alignment weight $\beta = 0.3$.

Iterative Protocol. The full `SHiFT` framework executes the EM loop (E-step $\leftrightarrow$ M-step) for a total of 3 iterations, which we empirically found sufficient for convergence.

C.3.3 Evaluation Protocol

We strictly follow the standard semi-supervised learning setting defined in `LLMNodeBed`.

- **Node Classification:** After the unsupervised representation learning, we freeze the GNN encoder and attach a linear classification head. We fine-tune this head using the official train/validation/test splits provided by the benchmark. We report the mean Accuracy and standard deviation across 5 random seeds.
- **Node Clustering:** Following [40, 37], we apply K-Means clustering directly on the learned node embeddings $\boldsymbol{Z}$. The number of clusters K is set to the ground-truth class count of each dataset. Performance is evaluated using Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI).

D List of LLM Prompts

We employ a systematic prompt engineering strategy to guide the LLM in inducing and refining the semantic taxonomy. All prompts are designed to output structured JSON objects to ensure parseability. In the templates below, placeholders enclosed in `{{ }}` denote dynamic content inserted during runtime. To optimize token usage, input document texts are truncated to the first 500 characters.

D.1 Taxonomy Induction (Cold-Start)

D.1.1 Concept Induction Prompt

This prompt corresponds to the E-step's Phase 3 (Section 4.2.1), used to generate a concept name and definition from a cluster of document embeddings.

System Prompt

You are an expert taxonomist assisting in organizing scientific literature.

User Prompt

Context: We are currently exploring sub-topics under the parent category: `{{ parent_info }}`.
Task: Below are `{{ num_docs }}` representative papers from a specific cluster of documents within this parent category. Your goal is to summarize the common research topic shared by these papers into a specific sub-concept.
Additionally, determine whether this sub-concept is semantically coherent and focused enough that it should NOT be further subdivided. If the papers clearly belong to a single, narrow topic with no distinct sub-directions, set "concept_split_needed" to false. If there appear to be multiple distinct sub-themes or methodological directions, set it to true.
Input Papers: `{{ documents_list }}`
Constraints: 1. The Concept Name should be concise (2-5 words) and standard in the field. 2. The Definition should be a single sentence explaining what this sub-concept focuses on, distinct from other sub-fields. 3. Ensure the concept strictly belongs to `{{ parent_info }}`.
Output Format (JSON): { "concept_name": "String", "definition": "String", "concept_split_needed": true or false }

D.1.2 Taxonomy Refinement Prompt

This prompt corresponds to the E-step's Phase 4 (Section 4.2.1), acting as an "Editor" to clean up the over-clustered candidates via MERGE or PRUNE operations.

System Prompt

You are a senior editor optimizing a hierarchical taxonomy.

User Prompt

Context: Under the parent category `{{ parent_info }}`, we identified `{{ num_candidates }}` candidate sub-concepts based on clustering. However, the clustering algorithm may have produced: 1. Duplicates/Synonyms (e.g., "GNN" and "Graph Neural Nets" should be MERGED). 2. Overly fragmented topics (e.g., "LSTM applications" and "RNN models" might be MERGED into "Recurrent Neural Networks"). 3. Outliers/Irrelevant topics (e.g., a Biology topic appearing in a CS list should be PRUNED).
Candidate Sub-concepts: ID | Name | Definition `{{ concept_list }}`
Task: Review the list and output a set of operations to refine this level of the taxonomy.
Allowed Operations: - MERGE: Combine IDs [id1, id2, ...] into a new, broader concept. Provide a new Name and Definition. - PRUNE: Remove ID [id] if it is clearly off-topic, too vague, or low quality. - KEEP: Keep ID [id] as is (or rename it slightly for clarity).
Output Format (JSON list): [{"op": "MERGE", "source_ids": [1, 5, 8], "new_name": "Graph Neural Networks", "new_def": "...", "reason": "Synonyms"}, {"op": "PRUNE", "target_id": 3, "reason": "Off-topic physics paper"}, {"op": "KEEP", "target_id": 2, "new_name": "Transformers"}]

D.2 Taxonomy Maintenance (Incremental Updates)

D.2.1 Merge Diagnosis Prompt

Used in the "Surgical Diagnosis" phase (Section 4.2.2) when sibling concepts exhibit high cosine similarity. It supports multi-way merging strategies.

System Prompt

You are an expert taxonomist optimizing a document taxonomy.

User Prompt

Context: Under the parent category `{{ parent_info }}`, we identified `{{ num_siblings }}` sibling concepts that appear semantically similar based on their vector representations.
Similar Concepts: `{{ concept_details }}`
Task: Analyze these concepts and determine the best merging strategy.
You can output ONE of the following strategies: Strategy A: MERGE ALL (Use when they are all synonyms or very closely related) Strategy B: PARTIAL MERGE (Use when there are subgroups of similar concepts) Strategy C: KEEP ALL SEPARATE (Use when they are genuinely distinct despite high similarity)
Consider: 1. Are they synonyms? (e.g., "GNN", "Graph Neural Nets", "Graph NN") 2. Are they closely related subtopics? (e.g., "LSTM", "GRU", "Recurrent Nets") 3. Are they distinct but related? (e.g., "Image Classification", "Object Detection")
Output Format (JSON): { "strategy": "MERGE_ALL" or "PARTIAL_MERGE" or "KEEP_SEPARATE", "reasoning": "Brief explanation of your decision", "operations": [...] }

D.2.2 Definition Refinement Prompt

Triggered when a cluster experiences significant member turnover (Concept Drift), asking the LLM to update the definition based on new core-set documents.

System Prompt

You are an expert taxonomist refining concept definitions.

User Prompt

Context: We have an existing concept whose member documents have changed significantly after embedding updates.
Current Concept: - Name: `{{ current_name }}` - Definition: `{{ current_def }}`
New Representative Papers: `{{ new_docs }}`
Task: Determine if the current definition still accurately describes these papers, or if it needs refinement.
Consider: 1. Does the old definition still capture the main theme of these papers? 2. Has the focus shifted slightly (e.g., from theory to applications)? 3. Should we keep the name but refine the definition?
Output Format (JSON): { "should_update": true or false, "reason": "Brief explanation", "new_name": "Refined name (can be same as old)", "new_definition": "Refined definition" }

D.2.3 Outlier Assignment Prompt

Used to re-assign "unassigned bucket" nodes to the most relevant existing concept based on semantic content.

System Prompt

You are an expert taxonomist assigning documents to concepts.

User Prompt

Context: We have a document that doesn't clearly belong to any existing cluster. We need to assign it to the most appropriate concept.
Document: `{{ document_text }}`
Candidate Concepts: `{{ candidate_list }}`
Task: Which concept does this document best fit into? Consider the core topic and methodology.
Output Format (JSON): { "best_match_id": 1 or 2 ... (or 0 if none fit), "reason": "Brief explanation" }

E Taxonomy Evaluation Protocols

E.1 Concept Coherence Calculation

To strictly quantify the semantic coherence of the induced taxonomy, we employ Normalized Pointwise Mutual Information (NPMI). The calculation proceeds in two steps:

1. Keyword Extraction via c-TF-IDF. We first aggregate all documents assigned to a concept c into a single "super-document". We then calculate the Class-based TF-IDF score for each word w in concept c:

$$W_{w,c} = \text{tf}_{w,c} \times \log\left(1 + \frac{N}{\text{df}_w}\right), \tag{37}$$

where $tf_{w,c}$ is the frequency of word w in concept c, N is the total number of concepts, and df_w is the number of concepts containing w. We select the top-10 words with the highest scores as the representative keyword set W_c.

2. NPMI Coherence Score. We measure the pairwise co-occurrence of words in W_c across the entire corpus using a sliding window. The NPMI for a pair (w_i, w_j) is defined as:

$$\text{NPMI}(w_i, w_j) = \frac{\log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}}{-\log P(w_i, w_j)}, \tag{38}$$

where $P(w)$ is the probability of word occurrence. The final score for the taxonomy is the average NPMI over all pairs in all concepts.

E.2 Human Evaluation Protocol

We recruited three CS graduate students (with backgrounds in NLP and graph learning) as annotators. The evaluation was conducted on the induced taxonomies of Cora, covering three taxonomy variants: the text-only baseline ($\mathcal{T}_{\text{w/o } \mathcal{G}}$), the cold-start initialization ($\mathcal{T}_{\text{init}}$), and the final `SHiFT` taxonomy ($\mathcal{T}_{\text{final}}$).

For each dataset and each taxonomy variant, we randomly sampled 20 concept nodes. Each annotator was presented with:

- The concept name and definition generated by the LLM.
- Five representative documents assigned to that concept (selected by the dual-level core-set sampling described in Sec. 4.2.1).
- The concept's parent node name and definition (to enable assessment of the parent-child relation).

Annotators were asked to rate each concept on the following three criteria using a 1–5 Likert scale (1 = strongly disagree, 5 = strongly agree):

- **Readability**: The concept name and definition are clear, concise, and well-defined.
- **Coherence**: The representative documents genuinely and consistently belong to this concept.
- **Hierarchy Validity**: The parent-child relationship between this concept and its parent is logically sound and follows an is-a relation.

Table 8: Hierarchical classification on Patent (%). Best results are in bold.

Method	Acc@L3	Acc@L2	Acc@L1	hF
MLP	64.32 ±0.58	73.65 ±0.65	81.32 ±0.49	73.45 ±0.42
GCN	70.91 ±0.52	78.52 ±0.33	83.84 ±0.18	77.85 ±0.39
TAPE	71.54 ±0.41	79.43 ±0.28	84.41 ±0.31	78.96 ±0.33
SHiFT	**74.18** ±0.44	**81.27** ±0.31	**85.73** ±0.25	**81.34** ±0.28

The three taxonomy variants were presented in randomized order, and annotators were blind to which variant they were evaluating. The annotation interface displayed the concept information in a unified format to prevent visual bias.

E.3 Taxonomy as Features Details

In the extrinsic evaluation (Sec 5.3), we construct the augmented feature input as follows:

1. **Path Extraction:** For a node v assigned to a leaf concept, we trace the path from the root $c_{root} \to \cdots \to c_{leaf}$.
2. **Text Linearization:** We concatenate the names and definitions of all concepts along the path to form $\mathcal{P}(v) =$ "[ROOT] $name_{root} : def_{root} \ldots$ [LEAF] $name_{leaf} : def_{leaf}$".
3. **Input Construction:** The final input is $S_{new} = S_{raw} \oplus$ [SEP] $\oplus \mathcal{P}(v)$.
4. **Encoding:** We encode S_{new} using the frozen PLM to obtain $X_{aug} \in \mathbb{R}^d$.

We then train a standard 2-layer MLP and a 2-layer GCN on X_{aug} using the cross-entropy loss.

F Evaluation on a Patent Graph with Verifiable Hierarchy

This section tests whether an induced taxonomy recovers a hierarchy that was assigned independently of the model. The flat-label benchmarks cannot answer that question on their own (Appendix C.1).

F.1 Dataset

We construct **Patent**, a text-attributed graph of 10,090 patents. Node text (title and abstract) and examiner citation links are taken from the public USPTO collection at https://github.com/opendata-ai/tr. Each node is annotated with its primary Cooperative Patent Classification (CPC) code from Google Patents Public Data (CC BY 4.0). CPC codes are assigned by USPTO patent examiners. Restricting to the primary code yields a strict 3-level tree on this graph: **L1** Section (7 classes), **L2** Class (55 classes), and **L3** Subclass (162 classes). Gold concept names are read from the official CPC Title List (USPTO/EPO, May 2026).[2] We use a temporal split, train ≤ 2013, validation 2014–2015, and test ≥ 2016, and release the processed graph with our code.

F.2 Hierarchical Classification

We compare SHiFT with MLP, GCN, and TAPE under the same encoder and training protocol as Section 5.1, using the temporal split above. Each method is scored at all three CPC levels. Hierarchical F-score follows Kiritchenko et al. [24]. For a test patent i, let Y_i be the gold subclass together with its ancestors, excluding the root, and let $\hat{Y}_i$ be the analogous set for the prediction. Micro-averaged hierarchical precision, recall, and F-score are

$$hP = \frac{\sum_i |\hat{Y}_i \cap Y_i|}{\sum_i |\hat{Y}_i|}, \qquad hR = \frac{\sum_i |\hat{Y}_i \cap Y_i|}{\sum_i |Y_i|}, \qquad hF = \frac{2\, hP\, hR}{hP + hR}. \tag{39}$$

A subclass error that still matches the true class or section receives partial credit, so hF is not a restatement of Acc@L3. Results over five seeds are in Table 8.

[2] https://www.cooperativepatentclassification.org

Table 9: Agreement between induced taxonomies and the official CPC tree on Patent.

Taxonomy	Precision	Recall	Cophenetic r
$\mathcal{T}_{\text{w/o } \mathcal{G}}$ (text-only)	0.7969	0.5160	0.2852
$\mathcal{T}_{\text{init}}$ (cold-start)	0.8814	0.5863	0.5603
$\mathcal{T}_{\text{final}}$ (Ours)	**0.9145**	**0.6399**	**0.6423**

The margin over the strongest flat baseline, TAPE, increases with granularity: $+1.3$ at L1, $+1.8$ at L2, and $+2.6$ at L3, with $+2.4$ on hF. Coarse section labels are easy for a strong text model; subclass labels are where coarse-to-fine structure helps. The hF gain is the same phenomenon measured along the ancestor chain rather than at a single cut of the tree. This is the pattern expected when the encoder uses the hierarchy, and it is not explained by graph smoothing alone: GCN already has the citation graph, and TAPE already has LLM-enhanced text.

F.3 Alignment with the CPC Tree

Classification shows that hierarchy-aware representations help when the label tree is the target. We next ask whether the taxonomy `SHiFT` induces, without reading CPC symbols, agrees with that tree. We compare the three taxonomies already used in Section 5.3: text-only $\mathcal{T}_{\text{w/o } \mathcal{G}}$ (PLM embeddings), cold-start $\mathcal{T}_{\text{init}}$ (after graph-contrastive warm-up), and $\mathcal{T}_{\text{final}}$ (after EM refinement).

Node-level semantic alignment. For an induced concept at depth $d \in \{1, 2, 3\}$, we mean-pool frozen RoBERTa embeddings of the title and abstract of patents assigned to that concept. CPC classes at the same level are embedded in the same way, using the patents that carry each code. Each induced concept retrieves its five nearest CPC classes by cosine similarity. We count a match when the nearest neighbor exceeds a threshold τ fixed before the comparison and shared by all three taxonomies. Precision is the fraction of induced concepts with a match. Recall is the fraction of CPC classes matched by at least one induced concept.

Document-level hierarchy recovery. Semantic match of concept names can succeed even when the tree geometry is wrong. For every pair of patents we compute the cophenetic distance under the induced tree and under the CPC tree, using the depth of their lowest common ancestor at the leaf level [53], and report the Pearson correlation between the two distance vectors. The score is high only when sibling and cousin relations agree, not merely when individual nodes have a nearby CPC title.

Table 9 supports three observations. (1) *The citation graph carries hierarchical signal that text alone misses.* $\mathcal{T}_{\text{w/o } \mathcal{G}}$ already matches many CPC titles (precision 0.80) but barely preserves pairwise tree distance ($r = 0.29$). Adding the patent graph in $\mathcal{T}_{\text{init}}$ nearly doubles cophenetic correlation ($0.29 \rightarrow 0.56$). Concepts induced from text can look locally coherent and still scramble which documents are siblings. (2) *EM refinement improves coverage and geometry together.* From $\mathcal{T}_{\text{init}}$ to $\mathcal{T}_{\text{final}}$, precision rises by 0.033, recall by 0.054, and cophenetic correlation by 0.082. The same ordering, text-only $<$ cold-start $<$ final, appears in NPMI and in human Hierarchy Validity on Cora, Citeseer, and WikiCS (Section 5.3). Those proxy metrics were not an artifact of missing labels: the same refinement moves the induced tree toward an external gold hierarchy. (3) *Full recovery of CPC is not the objective.* Recall 0.64 and $r = 0.64$ leave a real gap. `SHiFT` names concepts from the documents and chooses its own granularity; it does not emit USPTO symbols. The monotonic gain across all three metrics is still evidence that this graph contains a recoverable hierarchy, and that hierarchy-aware induction tracks it more closely than a graph-agnostic clustering of the same text.

G Efficiency and Scalability Analysis

We analyze the efficiency of `SHiFT` from three perspectives: theoretical complexity, empirical runtime comparison against baselines, and a detailed breakdown of LLM inference costs.

G.1 Complexity Analysis

E-Step (Taxonomy Induction). The cold-start initialization performs hierarchical K-Means clustering over n nodes with embedding dimension d. At each recursion level, clustering n nodes

Table 10: Efficiency comparison on ArXiv. Baseline results are from `LLMNodeBed` running on an **H100 (80 GB)**; `SHiFT` runs on a **V100 (32 GB)**. Time is reported in hours (h), minutes (m), or seconds (s).

Metric	GCN_{SE}	SenBERT	RoBERTa	ENGINE	GCN_{LLM}	TAPE	GraphGPT	LLaGA	LLM_{IT}	`SHiFT` **(Ours)**
GPU Mem. (GB)	4.79	9.98	46.10	7.08	7.36	46.10	60.41	41.94	69.13	**16.76**[†]
Total Time	51s	7m	41m	2.6h	1.4h	37.4h	7.8h	7.7h	36.3h	**5.5h**

[†] `SHiFT` runs on a 32 GB V100; methods with >40 GB memory require 80 GB A100/H100 GPUs.

into K micro-clusters costs $\mathcal{O}(nKd)$ per K-Means iteration. Core-set sampling via CCPR requires computing PageRank on induced subgraphs, which costs $\mathcal{O}(|\mathcal{E}|)$ in total across all clusters. LLM calls scale with the number of micro-clusters K_{total} across all recursion levels; since $K_{\text{total}} \ll n$ in practice, this step is decoupled from graph scale. For the incremental update (Mapping & Diagnosis), node-to-centroid assignment costs $\mathcal{O}(nKd)$, and LLM calls are triggered only for anomalous clusters, reducing to a small fraction of K_{total}.

M-Step (Representation Learning). The GNN forward pass costs $\mathcal{O}(|\mathcal{E}|d + nd^2)$ per layer, consistent with standard GCN complexity [23]. Constructing the partition-augmented adjacency $\boldsymbol{A}^+$ for TPA adds at most $\mathcal{O}(n^2/K)$ intra-cluster pairs on average, which is manageable for moderate K. TSA computes pairwise distances among K cluster centroids, costing $\mathcal{O}(K^2d)$. TCA evaluates path-wise distances for each node over L levels, costing $\mathcal{O}(nLd)$. Since $K \ll n$ and L is small (typically ≤ 4), both TSA and TCA introduce negligible overhead relative to the GNN backbone.

Overall, the dominant cost per EM iteration is $\mathcal{O}(|\mathcal{E}|d + nKd)$, which is comparable to a standard graph contrastive learning pipeline. The LLM cost is bounded by $\mathcal{O}(K_{\text{total}})$ calls per E-step, which decreases substantially after the cold-start phase.

G.2 Empirical Efficiency Comparison

We conduct a comprehensive efficiency analysis on the large-scale ArXiv dataset (169,343 nodes; 1,166,243 edges) and compare `SHiFT` against representative baselines from the `LLMNodeBed` [63].

Hardware Setup. All baseline results are reported on a single NVIDIA H100 (80GB) GPU. `SHiFT` is evaluated on a single NVIDIA Tesla V100 (32GB) GPU, which has considerably lower memory bandwidth and compute capacity. This hardware asymmetry makes the comparison conservative and favorable to the baselines.

Table 10 summarizes GPU memory usage and total training time. Since `SHiFT` uses LLM through an external API, its reported GPU memory reflects only the local graph-side computation, whereas some LLM-based baselines include the memory footprint of locally hosted LLMs. Therefore, Table 10 should be interpreted as a practical deployment-cost comparison under each method's original execution setting, rather than a hardware-parity comparison of total LLM serving cost. Under this interpretation, `SHiFT` has the following practical efficiency properties:

- **Low Local Memory Footprint.** `SHiFT` consumes only 16.76 GB of local GPU memory, significantly less than LLM-heavy methods such as GraphGPT (60.41 GB) and LLM_{IT} (69.13 GB). This makes `SHiFT` practical to run on widely available mid-range GPUs (e.g., V100, RTX 3090/4090).
- **Fast Wall-Clock Training.** `SHiFT` completes the full pipeline, including pretraining, taxonomy induction, and EM iterations, in approximately 5.5 hours. This is $6.8\times$ faster than TAPE (37.4 h) and $6.6\times$ faster than LLM_{IT} (36.3 h), validating our key design principle of decoupling heavy LLM reasoning (invoked sparingly) from lightweight GNN training (performed repeatedly).

G.3 Time Breakdown and LLM Cost Analysis

We further decompose the total runtime of `SHiFT` to quantify the benefit of our Mapping & Diagnosis strategy.

Training Time Composition. The total runtime ($\approx$ 5 h 29 m) is distributed as follows:

- **E-Step (Taxonomy Induction):** The cold-start initialization takes $\approx$ 1 h 19 m. Thanks to the incremental Mapping & Diagnosis design, subsequent E-steps are progressively cheaper: Update 1 takes $\approx$ 1 h 01 m, and Update 2 takes $\approx$ 59 m.
- **M-Step and Warm-up:** The remaining $\approx$ 2 h 10 m is spent on GCN training and contrastive pretraining, both of which are highly efficient.

Reduction in LLM API Calls. A key concern for LLM-integrated graph learning is inference cost. Table 11 tracks LLM API calls across E-step iterations. During cold-start initialization, 1,217 calls are required to construct the taxonomy from scratch. In subsequent updates, the Mapping & Diagnosis strategy (Section 4.2.2) restricts LLM invocation to anomalous clusters only, reducing calls to 285 in Update 1 (**76.6% reduction**) and 270 in Update 2. This confirms that the semantic structure stabilizes rapidly after initialization, incurring minimal marginal LLM cost in later iterations.

Table 11: Number of LLM API calls during E-steps on ArXiv.

Phase	LLM Calls	Reduction vs. Init.
Taxonomy Initialization	1,217	–
Incremental Update 1	285	$\downarrow$ 76.6%
Incremental Update 2	270	$\downarrow$ 77.8%

H Impact of LLM Selection

The core premise of SHiFT is leveraging the semantic reasoning capabilities of LLMs to induce interpretable taxonomies. To assess whether our framework relies on specific proprietary models or is robust across different backbones, we evaluate SHiFT using three different LLMs varying in scale and architecture: (1) **DeepSeek-V3** [7]: The default large-scale model used in our main experiments; (2) **Qwen3-32B** [66]: A powerful open-weights model with strong reasoning capabilities; (3) **Qwen3-8B** [66]: A lightweight model suitable for consumer-grade GPUs. We access these LLMs via the API provided by Alibaba Cloud. We compare these against a **w/o LLM** baseline, which represents the standard GCN trained on PLM features without any taxonomy-induced constraints.

Table 12: Node classification accuracy (%) with different LLM backbones.

LLM Used	Cora	Citeseer	WikiCS
DeepSeek-V3 (Default)	**84.83** ±0.58	**73.96** ±0.92	**82.40** ±0.48
Qwen3-32B	82.44 ±0.37	73.24 ±1.03	82.04 ±0.41
Qwen3-8B (Lightweight)	84.14 ±0.60	72.97 ±0.27	81.65 ±0.18
w/o LLM (GCN Backbone)	81.70 ±1.01	69.71 ±0.64	79.95 ±0.15

The results are presented in Table 12. We observe three key trends:

1. Robustness Across Architectures. SHiFT consistently outperforms the **w/o LLM** baseline regardless of the LLM backend used. Even with the smallest model (Qwen3-8B), we observe substantial gains (e.g., $+3.26\%$ on Citeseer compared to baseline). This demonstrates that the effectiveness of SHiFT stems from the methodology of aligning embeddings with a semantic hierarchy, rather than solely from the "knowledge injection" of a specific super-model.

2. Benefit of Stronger Reasoners. DeepSeek-V3 consistently yields the highest accuracy across all datasets (84.83% on Cora). This aligns with intuition: a more capable taxonomist (LLM) induces a cleaner, more logically consistent hierarchy (E-step), which provides higher-quality supervision signals for the GCN (M-step). For complex datasets like Citeseer, where community structures are sparse, the stronger reasoning capability of DeepSeek-V3 (73.96%) provides a noticeable edge over Qwen3-32B (73.24%) and 8B (72.97%).

3. Efficiency-Performance Trade-off. Notably, the lightweight Qwen3-8B delivers surprisingly competitive performance, coming within $\approx 0.7\%$ of DeepSeek-V3 on Cora and WikiCS. It even

marginally outperforms the larger 32B model on Cora (84.14% vs 82.44%), likely because the taxonomy of Cora is relatively simple, and the 8B model is less prone to over-complicating concept definitions. This finding is particularly encouraging for practical deployment: it suggests that SHiFT can achieve state-of-the-art results using locally deployable, open-source LLMs, significantly reducing inference costs and latency.

I Hyperparameter Sensitivity

I.1 Sensitivity to the Level Weighting Factor

In $\mathcal{L}_{tca}$, the weight $w_l = \lambda^l$ controls the relative importance assigned to each level of the hierarchy. We clarify two points here.

Motivation for $\lambda > 1$. When $\lambda = 1$, all hierarchy levels receive equal weight. This is suboptimal because coarse-grained concepts (e.g., "Computer Science") are shared across many nodes and provide limited discriminative signal, while fine-grained concepts (e.g., "Value Function Approximation") are far more informative for distinguishing individual nodes. Setting $\lambda > 1$ progressively emphasizes the deeper, more discriminative levels, reflecting this inherent coarse-to-fine structure of the taxonomy.

Sensitivity Study. To empirically justify our default choice of $\lambda = 1.5$, we conduct a sensitivity analysis by varying $\lambda \in \{1.0, 1.25, 1.5, 2.0\}$ on Cora and Citeseer. Results are reported in Table 13. Equal weighting ($\lambda = 1$) underperforms because the uniformly distributed loss allows the loose, less discriminative upper-level constraints to dilute the fine-grained alignment signal. Overly aggressive weighting ($\lambda = 2$) also degrades performance, as it disproportionately focuses on leaf-level concepts and neglects the coarse-level regularization that stabilizes the global geometry of the embedding space. $\lambda = 1.5$ strikes an effective balance between these two extremes and is adopted as the default throughout our experiments.

Table 13: Sensitivity of node classification accuracy (%) to the level weighting factor λ.

λ	Cora	Citeseer
1.0 (equal)	84.21	73.18
1.25	84.52	73.67
1.5 (default)	**84.83**	**73.96**
2.0	84.45	73.40

I.2 Sensitivity to the Taxonomy Depth

We investigate the impact of the maximum taxonomy depth L_{max} on node classification performance. As shown in Table 14, we observe a consistent trend where performance peaks at Depth 3 across the evaluated datasets (Cora, Citeseer, WikiCS).

We attribute this to a trade-off between semantic granularity and structural noise:

- **Depth 2 (Too Coarse):** The hierarchy lacks sufficient depth to capture fine-grained semantic distinctions (e.g., specific methods within a sub-field), limiting the discriminative power of the embeddings.
- **Depth 4 (Too Fine):** For these medium-scale academic graphs, Depth 4 often leads to over-fragmentation, where coherent concepts are artificially split into minute sub-clusters, introducing noise into the alignment objectives.

Therefore, we set $L_{max} = 3$ as the default configuration for these benchmarks.

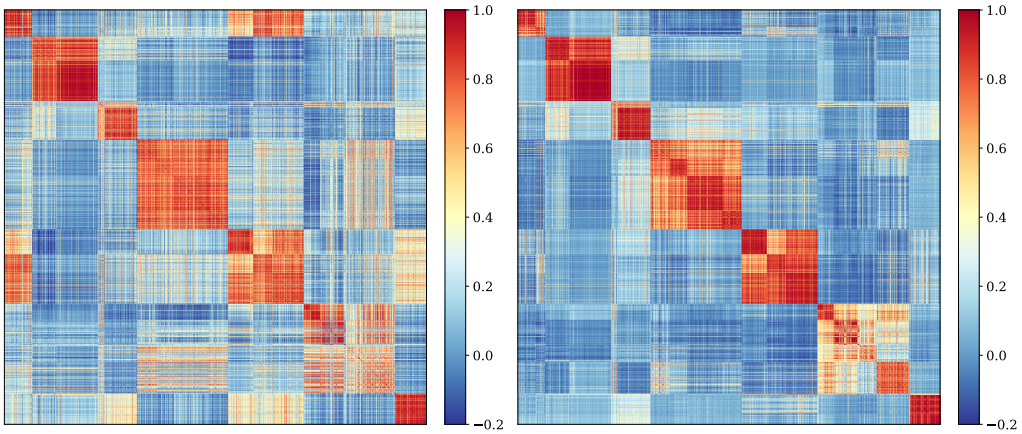
Note on Scalability. It is worth noting that the optimal depth is likely correlated with the scale and semantic diversity of the graph. While $L_{max} = 3$ suffices for the datasets evaluated here (2k-10k nodes), significantly larger or broader datasets (e.g., full-scale `ogbn-papers100M` [17] with millions of nodes) may inherently possess a deeper semantic structure. In such cases, increasing L_{max} (e.g., to 5 or 6) might be necessary to fully disentangle the complex topic hierarchy without over-fragmentation.

J Visual Analysis of Representations

To further investigate the geometry of the learned embedding space, we plot the heatmap of the pairwise cosine similarity matrix in Figure 3. The nodes are ordered based on the Depth-First Search

Table 14: Impact of maximum taxonomy depth (L_{max}) on node classification accuracy (%).

Depth	Cora	Citeseer	WikiCS
2	84.40 ±0.30	73.57 ±0.63	81.63 ±0.17
3 (Default)	**84.83 ±0.58**	**73.96 ±0.92**	**82.40 ±0.48**
4	84.20 ±0.20	72.70 ±0.82	81.85 ±0.65



(a) GCN Baseline (b) SHiFT (Ours)

Figure 3: Visualization of the pairwise cosine similarity matrix of node embeddings on the Cora dataset.

traversal of the taxonomy $\mathcal{T}$ induced by SHiFT. This ordering ensures that semantically related nodes (siblings and cousins) are placed adjacently.

Baseline Analysis. As observed in Figure 3(a), the baseline GCN produces a diffuse block pattern. While it captures some local community structures (indicated by reddish regions), it fails to reveal a clear hierarchical organization. The significant noise in the off-diagonal areas (blue regions containing yellow/orange streaks) suggests poor semantic disentanglement, where topologically connected but semantically distinct nodes are pulled too close due to the aggregation mechanism.

SHiFT **Analysis.** In contrast, SHiFT (Figure 3(b)) demonstrates a sharp hierarchical structure. We observe a "blocks-within-blocks" pattern:

- **Fine-grained Cohesion:** The intense red squares along the diagonal correspond to leaf concepts. The sharp boundaries indicate that the Taxonomy Partition Alignment successfully enforces strict local clustering.
- **Coarse-grained Structure:** Crucially, these small squares are grouped into larger, lighter red blocks. This reflects the Taxonomy Skeleton Alignment, which ensures that sibling concepts remain geometrically closer to each other than to concepts in distant branches.
- **Semantic Disentanglement:** The off-diagonal regions are cleanly separated (deep blue), confirming that SHiFT effectively distinguishes between disparate semantic branches (e.g., separating "Neural Networks" from "Probabilistic Methods").

K Qualitative Analysis of Taxonomies

To understand how our method uncovers the semantic hierarchy of text-attributed graphs, we conduct a detailed case study on the Cora dataset. We compare three variations of the induced taxonomy: (1) **PLM-only ($\mathcal{T}_{\text{w/o } \mathcal{G}}$):** Induced solely by clustering RoBERTa embeddings without graph structure; (2) **Structure-Aware Initialization ($\mathcal{T}_{\text{init}}$):** Induced after the graph contrastive pretraining warm-up; (3) SHiFT **Final ($\mathcal{T}_{\text{final}}$):** The final result after iterative EM updates. The visualizations of these taxonomies are presented in Figure 4, Figure 5, and Figure 6, respectively.

K.1 Structural Disentanglement vs. Semantic Redundancy

The most striking difference lies in the global topology. As shown in Figure 4, the PLM-only baseline fails to distinguish citation communities from lexical topics. It generates a generic "Machine Learning" child node under the root, which is uninformative for a dataset entirely composed of ML papers. More critically, it exhibits *structural hallucination*: "Neural Network Learning" appears twice, once as a child of "Machine Learning" and once as an independent root child. This confirms that without topological guidance, pure semantic similarity is insufficient to separate distinct research sub-fields (e.g., distinguishing "Rule Learning" papers from "Neural Network" papers, which use similar vocabulary but rarely cite each other).

In contrast, both $\mathcal{T}_{\text{init}}$ (Figure 5) and $\mathcal{T}_{\text{final}}$ (Figure 6) utilize the graph structure to pull the embedding space into topologically coherent regions. `SHiFT` successfully identifies exactly 7 top-level concepts, which map near-perfectly to the ground-truth classes of Cora, as detailed in Table 15. For instance, the vague "Machine Learning" cluster is replaced by specific communities like "Decision Tree Learning" (Rule Learning) and "Neural Network Modeling", effectively resolving the redundancy issue.

Table 15: Mapping between Ground-Truth classes and `SHiFT`-induced concepts.

Cora Ground Truth	`SHiFT` **Induced Concept** ($\mathcal{T}_{\text{final}}$)
Neural Networks	Neural Network Modeling
Rule Learning	Decision Tree Learning
Reinforcement Learning	RL with Function Approximation
Probabilistic Methods	Bayesian Graphical Models
Theory	Agnostic Learning Theory
Genetic Algorithms	Evolutionary Algorithms
Case Based	Case-Based Design Support

K.2 Hierarchy Depth and Concept Refinement

While the initialization step ($\mathcal{T}_{\text{init}}$) captures the skeleton, the iterative EM process ($\mathcal{T}_{\text{final}}$) significantly refines the semantic granularity and corrects logical inconsistencies.

Correction of Concept Drift. In the PLM-only tree, "Markov Chain Monte Carlo" is placed at Level 1, parallel to broad fields like "Evolutionary Algorithms". `SHiFT` corrects this by recognizing MCMC as a methodological tool. In $\mathcal{T}_{\text{final}}$, MCMC is correctly placed as a Level 3 sub-concept under "Bayesian Model Estimation", which itself sits under the "Bayesian Graphical Models" umbrella. This restores the logical *Field* $\rightarrow$ *Method* $\rightarrow$ *Technique* hierarchy.

Coarse-to-Fine Granularity. The final taxonomy demonstrates a sophisticated understanding of sub-fields. Consider the "Reinforcement Learning" branch:

- **Level 1 (Field):** *Reinforcement Learning with Function Approximation*, which correctly identifies the specific flavor of RL dominant in the Cora dataset.
- **Level 2 (Methodology):** *Dynamic Programming in RL*, which distinguishes model-based approaches from others.
- **Level 3 (Technique):** *Value Function Approximation*, a specific optimization technique within DP.

This deep hierarchy contrasts sharply with flat clustering methods, providing an interpretable roadmap of the scientific domain.

Semantic Precision. The LLM's role in definition generation shines in the "Theory" class. While the ground truth label is simply "Theory", `SHiFT` generates the concept "Agnostic Learning Theory", with a definition emphasizing "minimal assumptions about target functions". This accurately reflects the *Computational Learning Theory* (COLT) literature (e.g., PAC learning) prevalent in Cora, offering higher semantic density than human-annotated labels.

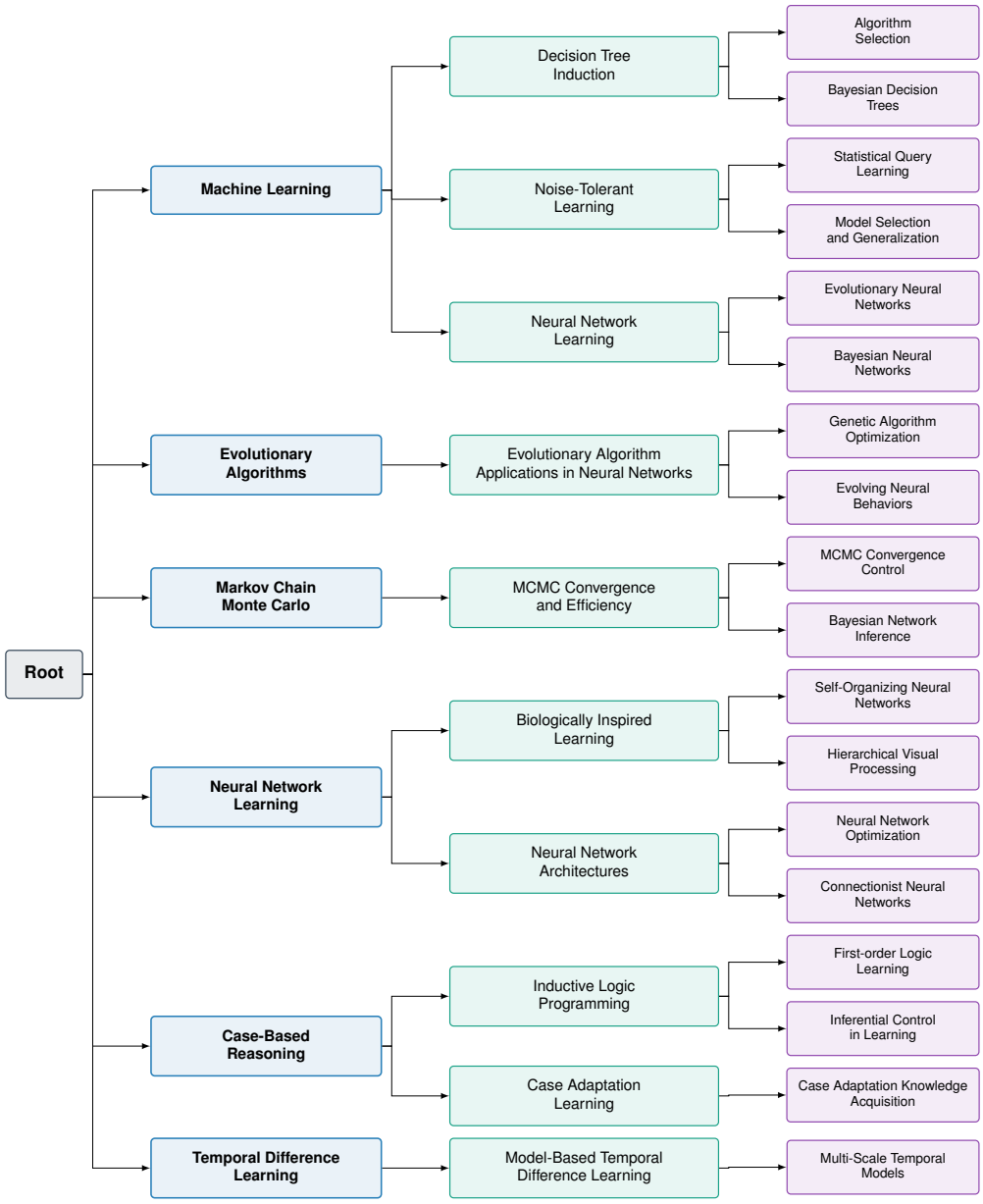


Figure 4: Cora taxonomy induced solely by PLM embeddings without graph structure ($\mathcal{T}_{\text{w/o } \mathcal{G}}$).

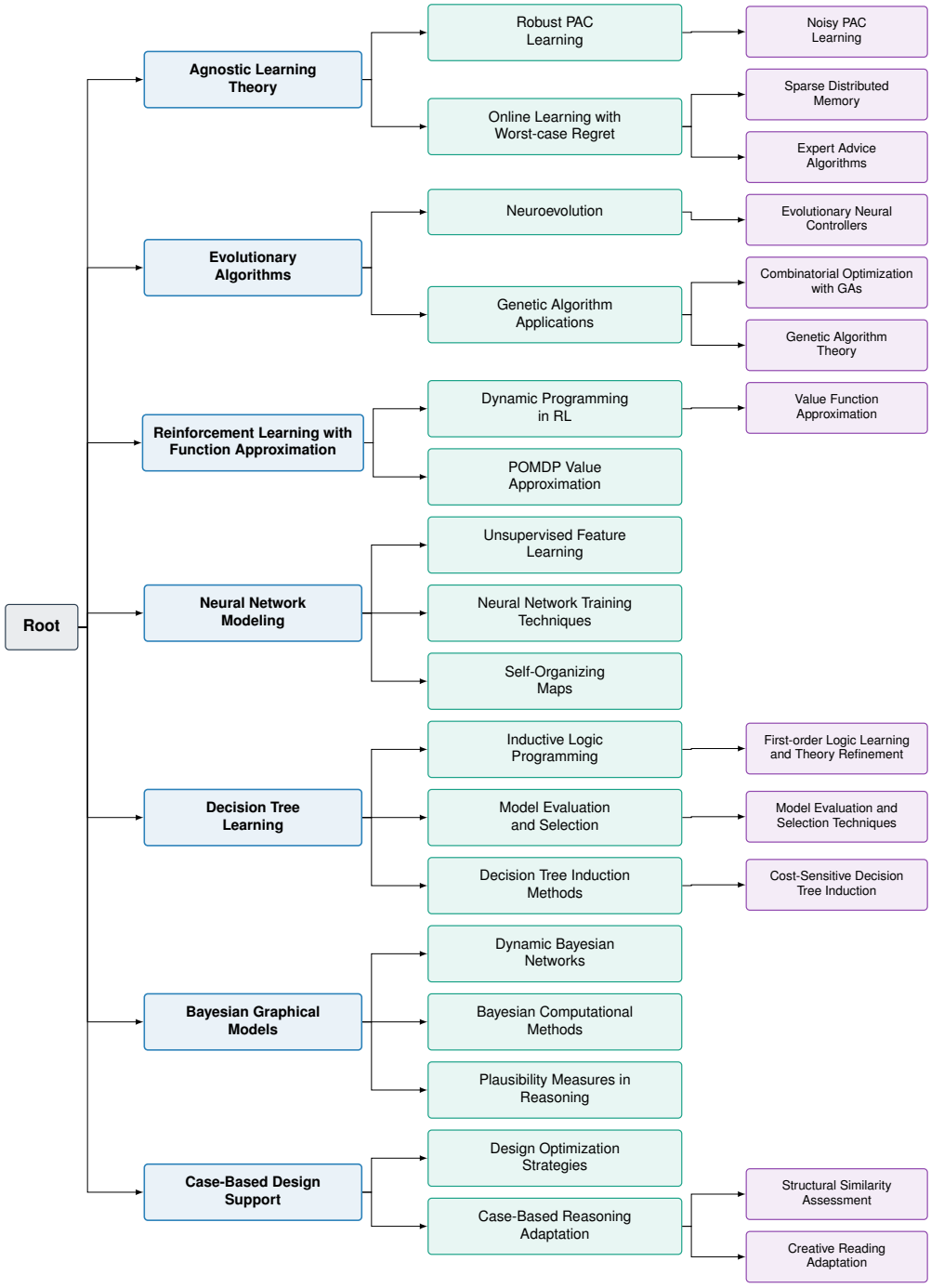


Figure 5: Cora taxonomy initialized after Graph Contrastive Pretraining ($\mathcal{T}_{\text{init}}$).

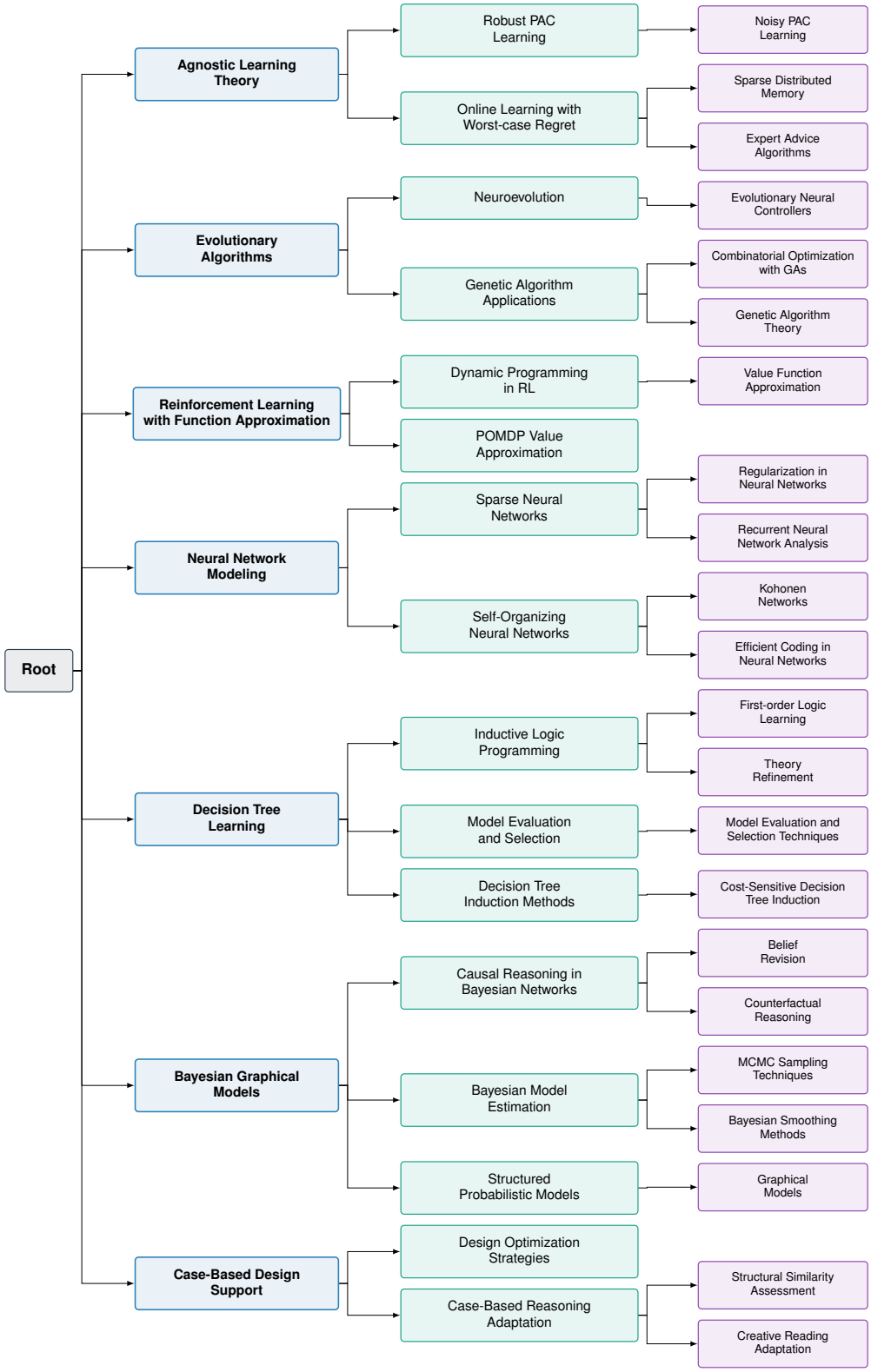


Figure 6: Final Cora taxonomy induced by SHiFT ($\mathcal{T}_{\text{final}}$).

L Miscellaneous

L.1 Limitations

Despite the promising results achieved by our framework in uncovering semantic hierarchies for text-attributed graphs, we acknowledge two primary limitations that define the scope of our current work and point toward future research directions.

Resource Consumption and Latency. First, the integration of LLMs introduces non-negligible computational overhead compared to traditional GNNs. Although our Mapping & Diagnosis strategy significantly reduces the frequency of LLM invocation, and our sensitivity analysis with Qwen3-8B (Appendix H) demonstrates that smaller, locally deployable models can yield competitive performance without relying on expensive proprietary APIs, the fundamental inference latency of LLMs remains a bottleneck. On extremely large-scale graphs with millions of nodes or in latency-sensitive applications, the cost and time required for the LLM to recursively cluster and define concepts may be prohibitive. Future work could explore distilling the taxonomic reasoning capabilities of large models into smaller, locally deployable student models to mitigate this bottleneck.

Scope of Hierarchical Evaluation. Second, most public TAG benchmarks still provide only flat labels. On Cora, Citeseer, ArXiv, WikiCS, and Computer we therefore rely on NPMI, human judgments, and downstream accuracy as proxies. Appendix F adds one direct test: a patent graph with a 3-level CPC taxonomy. Even there, `SHiFT` is not designed to reproduce the USPTO symbol inventory verbatim. It induces data-driven concepts, so CPC recall remains partial (0.64 for $\mathcal{T}_{\text{final}}$). A multi-domain suite of hierarchy-annotated TAGs is still needed before one can claim that explicit hierarchy modeling is necessary for every benchmark.

L.2 Ethical Considerations

To the best of our knowledge, the research presented in this paper conforms with the NeurIPS Code of Ethics. The work uses standard public benchmark datasets and a patent graph built from public USPTO records. It does not involve the collection of new personal data. It includes a small-scale, low-risk human evaluation for assessing taxonomy quality; this evaluation did not involve sensitive personal data and was conducted following the authors' institutional norms for such assessments. The proposed method is intended as a general-purpose research contribution and not for harmful applications such as surveillance, deception, or weaponization. We are not aware of any violations regarding privacy, consent, copyright, or responsible release, while acknowledging that pretrained language models and benchmark datasets may inherit existing biases that should be considered in downstream use.

L.3 Broader Impacts

This work is a foundational method for hierarchy-aware representation learning on text-attributed graphs. On the positive side, inducing explicit and human-readable taxonomies may improve the interpretability of graph learning systems and support applications such as scientific literature organization, knowledge discovery, and recommendation.

Potential negative impacts also exist. Because the induced hierarchy depends on input texts, graph structure, and LLM judgments, the method may inherit or amplify existing biases, leading to misleading or unfair semantic groupings. Moreover, when applied to sensitive relational data, explicit hierarchy induction may expose latent group structure and raise privacy concerns.

We therefore recommend human oversight, bias auditing of induced concepts, and cautious use on sensitive or high-stakes applications. The induced taxonomy should be treated as an assistive abstraction rather than ground truth.

L.4 Declaration of LLM Usage

Large Language Models (LLMs) are an essential component of our method. Specifically, in the E-step of our EM framework, we use an external LLM as a semantic reasoner to induce and refine the hierarchical taxonomy. The LLM is prompted to (i) generate concept names and definitions from

representative document sets, (ii) decide whether clusters require further splitting, and (iii) perform editing operations such as merging, pruning, and redefining concepts during incremental updates.

Importantly, the LLM is not used to generate node features, labels, or evaluation results. Instead, it serves as a structured semantic oracle whose outputs are explicitly integrated into the learning process via taxonomy construction and alignment objectives. All prompts, usage details, and implementation settings are provided in Appendix D to ensure transparency and reproducibility.