GRAPHTOKEN: A GENERAL GRAPH VOCABULARY FOR TEXT-ATTRIBUTED GRAPH FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph foundation models (GFMs) aim to enable generalizable graph learning across diverse tasks and domains, yet constructing transferable representation units for text-attributed graphs (TAGs) remains challenging due to the coupling of graph structure and textual semantics. Existing graph tokenization methods often derive tokens primarily from structural patterns or rely on continuous graph embeddings, leaving the learning of a shared discrete vocabulary over structure–text patterns underexplored. To address this challenge, we propose **GraphToken**, a motif-based discrete vocabulary learning framework for TAGs. For each target node, GraphToken samples multiple small rooted motif instances from its local neighborhood and jointly encodes their topology and node texts into root-conditioned motif representations. These representations are vector-quantized against a shared learnable codebook, where each codebook entry corresponds to a reusable *graph-token* shared by structurally and semantically related instances. A target node is consequently represented by a common set of graph-tokens that characterize its local structure–text context. Importantly, GraphToken decouples vocabulary induction from downstream prediction, allowing the learned tokens to be used by Graph Neural Networks (GNNs), Transformers, and Large Language Model (LLM)-based approaches alike, regardless of their architectural differences. Extensive experiments across graph domains, tasks, and model architectures demonstrate the effectiveness and generalizability of the learned GraphToken vocabulary. Anonymized code repository is available at https://anonymous.4open.science/r/GraphToken-15C3.

1 INTRODUCTION

Graph foundation models (GFMs) (Liu et al., 2023b; 2025; Wang et al., 2025b) seek representations that transfer across graph domains and downstream tasks (Shi et al., 2024; He et al., 2025; Xia et al., 2024). A key ingredient underlying such transfer in language models is a shared vocabulary (Conneau et al., 2020), which maps diverse inputs onto a finite set of reusable symbols and provides a common representational interface (Sennrich et al., 2016; Kudo & Richardson, 2018). An analogous interface for graphs, however, remains elusive. This challenge is particularly pronounced for text-attributed graphs (TAGs) (Zhao et al., 2023; Yan et al., 2023), where meaningful local patterns arise from the interplay between structures and textual semantics (He et al., 2023; Chen et al., 2024a). A transferable graph vocabulary must therefore abstract recurring structure–text patterns across graphs while retaining the ability to distinguish their semantic and structural roles (Li et al., 2026). This naturally raises the question of what should constitute the shared vocabulary of TAGs.

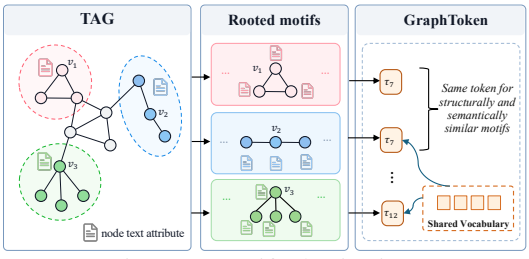


Figure 1: Motif tokenization.

A central difficulty lies in choosing graph units that are both informative and transferable. Node and edge attributes could transfer across graphs but expose only limited higher-order structure, whereas larger subgraphs capture richer context at the risk of becoming increasingly instance-specific (Li et al., 2025). Smaller-sized motifs provide a middle ground by capturing recurring local connectivity patterns at a compact scale (Milo et al., 2002; Alon, 2007; Benson et al., 2016). However, topology

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

alone often cannot capture the semantics of a TAG: isomorphic motifs carry different meanings when associated with different node texts, while similar textual content may play distinct structural roles (Yu et al., 2026). A meaningful vocabulary for TAGs must therefore identify recurring patterns in the joint space of structure and semantics, not treating either modality in isolation (Yan et al., 2023).

Existing graph representation and tokenization methods have explored reusable structural patterns, graph embeddings, and language-aligned tokens (Wang et al., 2024; Chen et al., 2024a; Tang et al., 2024). However, learning graph representations and learning a graph vocabulary are fundamentally different objectives. Representation learning maps each local graph instance to a continuous, instance-specific embedding, but does not explicitly determine which instances should share a common reusable identity (Qiu et al., 2020; You et al., 2020; Hou et al., 2022). At the other extreme, assigning a distinct token to every exact combination of topology and textual attributes yields a vast yet fragmented vocabulary with limited sharing across instances (Abolhasani et al., 2026). A transferable graph vocabulary (Galkin et al., 2024) must therefore go beyond representing local structure–text patterns and map related instances into a finite set of shared discrete identities.

To address these challenges, we propose GraphToken, a motif-based framework for learning a discrete vocabulary over TAGs. As illustrated in Figure 1, GraphToken decomposes a TAG into rooted local motifs and organizes their recurring structure-text patterns into a shared discrete vocabulary. GraphToken jointly encodes motif connectivity and node semantics into a common representation space, which is then quantized against a shared learnable codebook (van den Oord et al., 2017) to induce discrete identities, termed *graph-token* in our work. Structurally and semantically related motif instances can therefore share the same graph-token, suppressing instance-specific variation while preserving distinctions relevant to graph-based tasks. By coupling joint motif representation with codebook-based quantization, GraphToken provides a systematic mechanism for inducing transferable discrete graph-tokens over TAGs.

Beyond vocabulary construction, GraphToken establishes a shared interface between discrete graph-tokens and downstream architectures. For each target node, its local context is represented by a collection of recurring graph-tokens, yielding a vocabulary-grounded characterization of the neighborhood. Importantly, this token representation is defined independently of the downstream backbone. The same token space can therefore be consumed by diverse graph learning architectures, including GNNs, Transformers, and LLM-based methods, through lightweight aggregation or projection that adapt the token embeddings to their respective representation spaces. By decoupling graph tokenization from backbone design, GraphToken enables a single learned vocabulary to serve as a common representational foundation across different graph learning paradigms.

The main contributions of this work are summarized as follows. (1) We introduce **GraphToken**, a motif-level vocabulary learning framework that induces transferable discrete graph identities from joint structure–text subgraph representations. (2) We develop a backbone-independent tokenization framework that enables the learned graph vocabulary to interface with GNNs, Transformers, and LLMs. (3) Extensive experiments across graph domains, downstream tasks, and model architectures demonstrate the effectiveness and transferability of the learned vocabulary.

2 Related Work

Graph Foundation Models. GFMs seek transferable graph representations and learning principles that generalize across domains and downstream tasks (Liu et al., 2023b; 2025). GCOPE (Zhao et al., 2024) coordinates multiple source graphs to capture shared cross-domain knowledge, while MDGPT (Yu et al., 2024) introduces domain tokens and prompt-based adaptation for graphs with mismatched feature spaces. MDGFM (Wang et al., 2025a) aligns feature and topology distributions across domains, and SAMGPT (Yu et al., 2025) uses structure tokens and prompts to harmonize transferable structural patterns. Language-based formulations provide another route to transfer. OFA (Liu et al., 2023a) unifies graph entities and tasks through textual representations, UniGraph (He et al., 2025) couples language and graph encoders for cross-domain learning, GraphCLIP (Zhu et al., 2025b) promotes zero- and few-shot transfer through graph-text contrastive alignment, and GOFA (Kong et al., 2025) integrates graph reasoning with generative language modeling. OpenGraph (Xia et al., 2024) instead pursues generalization through unified graph tokenization and scalable Transformer pre-training. Despite their different formulations, these methods largely pursue transfer through representation alignment, prompting, pre-training objectives, or general-purpose architectures.

GraphToken instead focuses on the transferable units themselves, learning a finite vocabulary of shared discrete structure-text patterns that are reusable across graphs and downstream models.

Graph Vocabulary Learning. Graph tokens have increasingly served as an interface for translating graph structure into the representation space of sequence models. Representative approaches such as LLaGA (Chen et al., 2024a), GraphGPT (Tang et al., 2024), and RGLM (Zhang et al., 2026) align graph-derived representations with LLM embedding spaces through structural encoding, text–graph grounding, or reconstructive instruction tuning. In this paradigm, graph tokens function primarily as continuous, model-compatible carriers of structural information. Their role is therefore to bridge graph encoders and downstream language models, rather than to define reusable discrete identities within a shared graph vocabulary. A more closely related line of work seeks to construct reusable graph tokenizers or vocabularies that support transfer across graphs and tasks. Structural approaches derive shared units directly from recurring graph patterns, including computation trees in GFT (Wang et al., 2024) and tree–cycle vocabularies in RiemannGFM (Sun et al., 2025). OpenGraph (Xia et al., 2024) extends this idea toward a unified tokenizer capable of accommodating graphs with diverse structural characteristics within a scalable Transformer framework. More recent methods incorporate semantic information into vocabulary construction, with PromptGFM introducing language-based graph tokens for TAG generalization and GRAVER learning generative graph vocabularies for robust adaptation (Zhu et al., 2025a; Yuan et al., 2025). Complementary to these approaches, Guo et al. (2026) explores a BPE-inspired tokenizer that progressively composes structural units into higher-level graph tokens.

Motif-Based Discrete Representation. Graph motifs provide a compact abstraction of recurring higher-order connectivity and have been widely incorporated into graph kernels and neural architectures as structural primitives (Shervashidze et al., 2011; Sankar et al., 2017; Chen et al., 2024b). While these approaches establish small subgraphs as effective carriers of local structural regularities, motif identity is typically prescribed by topology, leaving the semantic variation among motif instances outside the token definition. GraphToken further connects this structural abstraction with neural discrete representation learning, where vector quantization induces shared latent identities over continuous representations (van den Oord et al., 2017). Our key distinction lies in the granularity and semantics of the representation being discretized. GraphToken instead casts motif identity as a learned partition of a joint structure–semantic space, allowing recurring local regularities to be abstracted into shared discrete units beyond topology alone.

3 PRELIMINARIES

In this section, we introduce key concepts and define the problem of graph vocabulary learning.

Text-Attributed Graphs. A text-attributed graph (TAG) is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V}$ and $\mathcal{E}$ are the node and edge sets, and $\mathcal{T} = \{t_v \mid v \in \mathcal{V}\}$ contains the node-level textual attributes. We obtain continuous node features as $\mathbf{x}_v = f_{\text{text}}(t_v)$ using a text encoder f_{text}. We consider a collection of TAGs $\mathcal{D} = \{\mathcal{G}_1, \ldots, \mathcal{G}_N\}$ that may vary in topology, scale, semantics, and domain.

Motif Instances. A motif instance in $\mathcal{G}$ is a subgraph $m = (\mathcal{V}_m, \mathcal{E}_m, \mathcal{T}_m)$, with $\mathcal{V}_m \subseteq \mathcal{V}$ and $\mathcal{E}_m \subseteq \mathcal{E}$ that matches a given motif pattern, and associated textual attributes $\mathcal{T}_m = \{t_u \mid u \in \mathcal{V}_m\}$ inherited from $\mathcal{G}$. For a target node v, let $\mathcal{M}_v$ denote its associated motif instances. In practice, $\mathcal{M}_v$ can be a sampled subset for efficiency.

Graph Vocabulary Learning. We learn a finite graph vocabulary $\{\tau_1, \ldots, \tau_K\}$, where each τ_k represents a reusable discrete identity, termed *graph-token*, shared across structurally and semantically related motif instances. A graph tokenizer assigns each motif instance m to a token τ_{c_m}, where $c_m \in \{1, \ldots, K\}$ is m's corresponding graph-token ID. This allows distinct motif instances to share vocabulary identities. For a target node v, it can be represented by a token *multiset* associated with its motif instances, $\{\tau_{c_m} \mid m \in \mathcal{M}_v\}$, which preserves the frequency information of recurring local structure-text patterns.

4 METHODOLOGY

In this section, we present the methodology of our proposed GraphToken solution. We begin by providing an overview of GraphToken, followed by a detailed development of its components.

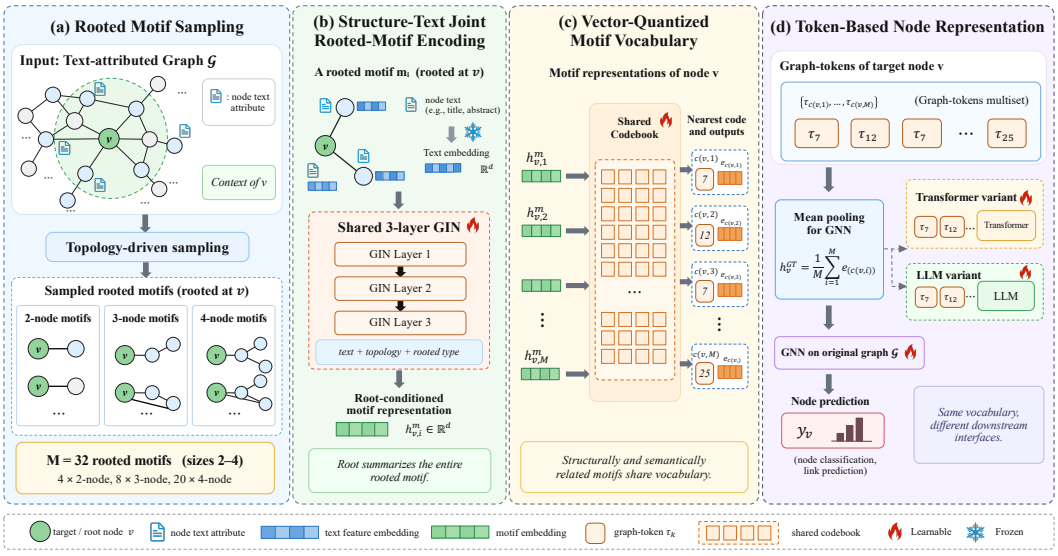


Figure 2: Overview of the GraphToken framework.

4.1 Overview

As shown in Figure 2, the GraphToken learns a shared discrete vocabulary over local structure–text patterns in TAGs. (a) Given an input TAG, GraphToken samples multiple rooted motif instances from the local neighborhood of each target node using topology-driven sampling. (b) A shared root-conditioned encoder jointly integrates motif topology and node text to obtain a representation of each rooted motif instance. (c) Vector quantization assigns these representations to their nearest prototypes in a shared vocabulary, thereby inducing reusable discrete motif identities. (d) The resulting graph-tokens form a compositional representation of the target node and can be directly consumed by different downstream backbones, including GNNs, Transformers, and LLMs.

4.2 Rooted Motif Sampling

GraphToken derives its vocabulary from compact rooted subgraphs sampled directly from the local graph topology. For each target node v, as shown in Figure 2 (a), we independently sample a set of M rooted motif instances, $\mathcal{M}_v = \{m_{v,1}, m_{v,2}, \ldots, m_{v,M}\}$, where each motif $m_{v,i}$ is rooted at v and contains at most S nodes. More specifically,

$$m_{v,i} = (\mathcal{V}_{v,i}, \mathcal{E}_{v,i}, \mathcal{T}_{v,i}, v)\,, \tag{1}$$

where v denotes the root, $\mathcal{V}_{v,i} \subseteq \mathcal{V}$ is the sampled node set and $|\mathcal{V}_{v,i}| \leq S$, $\mathcal{E}_{v,i} \subseteq \mathcal{E}$ denotes the corresponding induced edges from the original graph, and $\mathcal{T}_{v,i} = \{t_u \mid u \in \mathcal{V}_{v,i}\}$ contains the textual attributes of the sampled nodes.

Each rooted motif is sampled by stochastically expanding from the target node over its local topology until the prescribed motif size is reached or no further expansion is possible. Full sampling details are provided in Appendix A.4.3. We consider motifs of two to four nodes and use $M = 32$ sampling trials per target node, with 4, 8, 20 trials for size-two, -three, and -four rooted motifs, respectively. Trials are independent and prohibit repeated nodes within an instance, while duplicate motifs across trials are allowed. Because sampling is conditioned only on local topology rather than motif type or node labels, diverse configurations—including paths, stars, triangles, and cycles—emerge naturally from the neighborhood of the root.

4.3 Structure-Text Joint Rooted-Motif Encoding

The sampled rooted motifs capture local topology, but topology alone is insufficient for text-attributed graphs. As shown in Figure 2 (b), we therefore encode each rooted motif using both its connectivity and the textual attributes of its nodes.

For each node u, let $\mathbf{x}_u = f_{\text{text}}(t_u) \in \mathbb{R}^{d_x}$ denote its text representation obtained from a text encoder f_{text}. For a rooted motif $m_{v,i}$, the corresponding node representations are collected as

$\mathbf{X}_{v,i} = [\mathbf{x}_u]_{u \in \mathcal{V}_{v,i}}$. In addition to the local adjacency structure, each sampled motif is further characterized by a rooted structural type $\rho_{v,i} \in \{0, \ldots, R-1\}$, determined by its rooted structural equivalence class to capture both the local connectivity and structural role of the root (see details in Appendix A.4.4). The motif encoder therefore receives the node textual representations, the local connectivity, and the rooted structural type as complementary inputs, which are then passed into a GNN. We employ a L-layer Graph Isomorphism Network (GIN) (Xu et al., 2019) as the motif encoder shared across all rooted motifs, as follows.

$$\mathbf{H}_{v,i}^{(L)} = \mathrm{GIN}_\theta \left(\mathbf{X}_{v,i}, \mathcal{E}_{v,i}, \rho_{v,i} \right), \tag{2}$$

where $\mathbf{H}_{v,i}^{(L)}$ contains the contextualized representations of all nodes in the rooted motif $m_{v,i}$.

The GIN propagates node semantic information according to the sampled motif topology. At each message-passing layer, the representation of a node depends on both its own semantic state and those of its neighbors in the motif. Here, we use three GIN layers ($L = 3$) in our implementation. Since each rooted motif contains at most four nodes, any node in a connected motif is at most three hops away from the root. The root representation therefore has a receptive field covering the entire motif after three rounds of message passing.

We define the resulting rooted-motif representation as $\mathbf{h}_{v,i} = \mathbf{H}_{v,i}^{(L)}[v]$, i.e., the final hidden state of the root node v. Although $\mathbf{h}_{v,i}$ is read out at the root, it should not be interpreted as an isolated node representation. Instead, it contains information propagated from all nodes in $m_{v,i}$, making it a root-conditioned representation of the *entire* rooted motif. Consequently, motifs with similar textual attributes but different connectivity can receive different representations, while motifs with the same topology can be distinguished by their textual semantics.

4.4 Graph-tokens: Vector-Quantized Motif Vocabulary

As shown in Figure 2 (c), to transform continuous motif representations into reusable vocabulary units, we introduce a shared learnable codebook $\mathcal{C} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_K\}$, where each $\mathbf{e}_k \in \mathbb{R}^d$ is the embedding associated with the k-th discrete *graph-token* τ_k.

Given the continuous representation $\mathbf{h}_{v,i}$ of a rooted motif $m_{v,i}$, we assign it to the nearest codebook entry (van den Oord et al., 2017):

$$c(v,i) = \arg \min_{k \in \{1,\ldots,K\}} \|\mathbf{h}_{v,i} - \mathbf{e}_k\|_2^2, \tag{3}$$

where $c(v,i) \in \{1, \ldots, K\}$ is an index denoting the assigned *graph-token ID* for $m_{v,i}$. The rooted motif is therefore represented by the discrete graph-token $\tau_{c(v,i)}$, with $\mathbf{e}_{c(v,i)}$ as its corresponding embedding. This quantization maps instance-specific continuous motif representations to shared discrete identities. In particular, two distinct rooted motifs $m_{v,i} \neq m_{u,j}$ share the same graph-token if $c(v,i) = c(u,j)$, when their learned structure–text representations map to the same codebook prototype. This allows motifs to share a graph-token without requiring exact matches in nodes, textual content, or topology. Instead, token identity emerges from proximity in the joint structure–semantic space. The motif encoder and discrete vocabulary are jointly optimized through vector quantization, with a single shared vocabulary spanning all sampled motifs and target nodes.

As shown in Figure 2 (d), each target node v is represented by a *multiset* of graph-tokens assigned to its M sampled rooted motifs, $\{\tau_{c(v,1)}, \ldots, \tau_{c(v,M)}\}$. This is a multiset with repeated token assignments to retain frequency information across motifs rooted at v. For neural computation, each graph-token $\tau_{c(v,i)}$ is represented by its associated codebook embedding $\mathbf{e}_{c(v,i)}$. These embeddings are then aggregated or contextualized according to the downstream architecture.

4.5 Consumption of Graph-tokens

The learned graph-token vocabulary is decoupled from the choice of downstream predictor, which can be flexible across GNN-, Transformer-, and LLM-based backbones. In other words, these variants share a common graph-tokenization layer (with the same rooted-motif sampler, motif encoder, and discrete codebook), but differ in how the resulting graph-tokens are consumed for downstream tasks across different learning paradigms.

GNN-based consumption. For the GNN variant, we aggregate the embeddings of the graph-tokens assigned to node v by mean pooling:

$$\mathbf{h}_v^{\mathrm{GT}} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{e}_{c(v,i)}. \tag{4}$$

After obtaining $\mathbf{h}_v^{\mathrm{GT}}$ for all nodes, we use these vocabulary-level representations as node features on the original graph and apply a downstream GNN:

$$\mathbf{H}^{(\ell+1)} = \mathrm{GNN}^{(\ell)}\left(\mathbf{H}^{(\ell)}, \mathcal{E}\right), \qquad \mathbf{H}^{(0)} = \left[\mathbf{h}_v^{\mathrm{GT}}\right]_{v\in\mathcal{V}}. \tag{5}$$

For node classification and other tasks, the final node representation is further passed to their respective task head. The GNN variant therefore operates at two distinct structural levels. The shared GIN acts within each rooted motif to construct local graph-tokens, whereas the downstream GNN operates on the original graph to propagate the resulting graph-token representations.

Transformer-based consumption. The Transformer variant retains the individual graph-token embeddings rather than aggregating them beforehand. For each target node v, we prepend a shared learnable summary token $\mathbf{h}_v^{\mathrm{S}}$ and jointly contextualize it with the M graph-token embeddings. The output at the summary position is used as the target-node representation:

$$\mathbf{h}_v^{\mathrm{Trans}} = \mathrm{Transformer}\left(\left[\mathbf{h}_v^{\mathrm{S}}; \mathbf{e}_{c(v,1)}; \ldots; \mathbf{e}_{c(v,M)}\right]\right)[0]. \tag{6}$$

This formulation allows the Transformer to model interactions among graph-tokens derived from different rooted motifs of v, rather than compressing them through a fixed aggregation operator.

LLM-based consumption. For the LLM variant, we map each graph-token embedding into the language-model embedding space through a lightweight learnable projector $P_{\mathrm{LLM}} : \mathbb{R}^d \to \mathbb{R}^{d_{\mathrm{LLM}}}$. Given a task-specific prompt $\mathcal{P}_v$, the projected graph-token embeddings are inserted as graph-conditioned soft tokens:

$$\mathbf{O}_v = \mathrm{LLM}\left(\mathcal{P}_v, P_{\mathrm{LLM}}\left(\mathbf{e}_{c(v,1)}\right), \ldots, P_{\mathrm{LLM}}\left(\mathbf{e}_{c(v,M)}\right)\right). \tag{7}$$

5 Experiments

5.1 Experiment Setup

Datasets. To evaluate the multi-dataset generalization ability of GraphToken, we use ten benchmark graph datasets from three domains: citation networks, including Cora (McCallum et al., 2000), PubMed (Sen et al., 2008), ogbn-arxiv (arXiv) (Hu et al., 2020), and tape-arxiv23 (arXiv23) (He et al., 2023); e-commerce networks, including Children (Yan et al., 2023), History (Yan et al., 2023), Computers (Yan et al., 2023), Photo (Yan et al., 2023), and Fitness (Yan et al., 2023); and the social network Reddit (Huang et al., 2024). Following prior work, we adopt accuracy (Acc) and macro-F1 (F1) as evaluation metrics. More dataset details are provided in Appendix A.1.

Baselines. We compare GraphToken with four groups of representative baselines: *GNNs*, including GCN (Kipf & Welling, 2016), GAT (Veličković et al., 2017), and GraphSAGE (Hamilton et al., 2017); *pretrained graph models*, including GraphMAE (Hou et al., 2022), OFA (Liu et al., 2023a), UniGraph (He et al., 2025), and GraphCLIP (Zhu et al., 2025b); *graph–language models*, including GraphText (Zhao et al., 2023), LLaGA (Chen et al., 2024a), RGLM (Zhang et al., 2026), and GOFA (Kong et al., 2025); and *token/pattern models*, including GFT (Wang et al., 2024), OpenGraph (Xia et al., 2024), GRAVER (Yuan et al., 2025), and GPM (Wang et al., 2025c). Further details on the baseline methods, computational resources, and implementation settings are provided in Appendices A.2, A.3, and A.4, respectively.

5.2 Main Results

Performance on Node Classification. We evaluate GraphToken on supervised node classification across multiple TAGs. As shown in Table 1, GraphToken achieves competitive performance across datasets and model families, with the LLM-based variant generally providing the strongest results

Table 1: Node classification. Vacant positions ("-") indicate that the models ran out of memory on large graphs. Results are averaged over five runs, with the best in **bold**.

Methods	Cora		PubMed		arXiv		arXiv23		Reddit	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
GCN	81.88±0.43	72.15±0.41	83.55±1.01	82.89±0.98	71.01±0.08	44.74±0.30	67.98±0.42	24.86±0.67	52.65±4.54	40.72±9.17
GAT	82.62±0.43	74.84±1.56	83.33±0.09	82.46±0.12	71.46±0.10	47.47±0.22	68.08±0.09	27.56±0.32	54.21±3.52	46.70±10.09
GraphSAGE	81.92±0.87	74.35±2.21	82.54±2.42	82.43±2.14	71.97±0.10	45.10±0.44	68.80±0.65	28.04±0.61	54.94±2.98	53.65±4.13
GraphMAE	83.03±0.26	81.41±0.48	83.56±0.16	83.00±0.15	72.60±0.14	53.97±0.09	70.07±2.49	42.03±6.35	56.45±0.27	56.36±0.23
OFA	83.43±1.05	81.04±1.28	89.84±0.24	89.29±0.25	68.76±0.37	50.47±0.75	69.50±0.19	46.01±1.41	61.07±0.26	60.92±0.23
UniGraph	84.38±1.65	83.15±2.20	88.64±0.61	88.11±0.64	71.63±0.62	54.17±0.66	70.73±0.67	48.69±0.62	57.79±5.77	53.67±11.55
GraphCLIP	83.76±1.21	82.49±1.60	87.44±0.68	87.24±0.85	64.21±5.66	36.63±6.33	55.51±7.53	12.50±5.92	67.65±0.58	67.53±0.56
GraphText	83.21±0.26	81.61±0.36	83.19±1.06	82.72±1.02	–	–	69.58±1.04	43.26±1.02	58.77±0.19	56.18±0.36
LLaGA	83.39±0.52	81.33±0.68	89.10±0.36	88.98±0.33	73.06±0.06	53.13±0.16	68.68±0.01	33.25±0.14	65.62±0.15	65.41±0.15
RGLM	83.39±0.00	81.04±0.00	89.68±0.01	89.61±0.03	74.36±0.01	**56.40±0.01**	74.26±0.01	**56.40±0.02**	67.29±0.02	66.97±0.02
GOFA	83.03±1.44	80.74±1.35	**92.84±0.04**	**92.21±0.07**	75.28±0.49	54.73±0.92	75.91±1.15	55.54±2.45	54.16±4.88	46.77±10.74
GFT	84.72±1.08	83.42±1.18	90.06±0.31	89.66±0.41	72.22±0.26	48.32±0.40	70.79±0.21	43.25±2.23	59.37±0.35	58.84±0.94
GRAVER	74.54±3.78	73.62±3.33	74.19±3.04	73.89±3.00	52.83±3.03	36.43±2.30	46.75±1.95	32.83±0.61	57.97±1.41	57.93±1.48
OpenGraph	82.10±3.51	81.14±3.53	76.57±1.64	74.01±0.92	57.52±4.87	33.29±3.21	32.46±1.40	8.87±0.23	58.00±1.40	55.76±2.49
GPM	84.94±1.00	83.33±0.93	87.18±0.22	86.95±0.25	74.01±0.30	54.80±0.40	71.70±0.19	49.24±0.29	62.61±0.20	62.09±0.46
GraphToken$_{\text{GNN}}$	88.01±0.98	86.39±1.64	87.17±0.93	86.56±0.98	71.89±0.23	50.56±0.27	69.79±0.32	35.21±2.46	63.51±0.31	63.04±0.89
GraphToken$_{\text{Transf}}$	88.19±1.03	86.77±1.19	89.02±0.27	88.66±0.24	73.31±0.22	52.46±1.18	70.61±0.32	38.06±3.03	60.37±8.94	54.79±18.56
GraphToken$_{\text{LLM}}$	**89.12±0.76**	**87.63±0.91**	92.36±0.28	91.88±0.34	**75.46±0.27**	56.18±0.45	**76.08±0.39**	55.92±0.81	**67.82±0.31**	**67.68±0.38**
Methods	Children		History		Computers		Photo		Fitness	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
GCN	50.57±0.25	28.71±0.78	70.23±3.87	15.92±2.54	81.34±0.45	62.23±4.39	82.96±0.25	71.71±1.37	85.89±0.92	66.77±3.75
GAT	51.39±0.20	32.69±0.46	73.94±5.14	18.08±3.03	83.30±0.20	67.60±0.15	84.46±0.07	78.28±0.19	87.30±0.26	70.36±0.59
GraphSAGE	52.05±0.13	33.27±1.17	74.75±9.25	19.40±6.73	84.26±0.22	68.22±0.19	84.59±0.07	77.02±0.39	87.25±0.22	72.81±0.48
GraphMAE	53.95±0.13	44.33±0.14	73.29±2.06	17.84±0.91	84.75±2.63	73.98±3.56	84.55±0.70	78.22±1.69	90.62±0.17	83.55±0.37
OFA	54.92±0.37	43.03±0.67	84.29±0.26	51.91±1.56	89.37±0.18	84.99±0.61	86.27±0.13	82.00±0.30	86.14±0.27	82.00±0.42
UniGraph	56.59±0.62	45.06±0.49	83.68±0.39	51.34±1.29	89.99±0.16	85.09±0.70	87.06±0.14	82.76±0.29	91.86±0.05	85.98±0.22
GraphCLIP	31.88±0.88	3.14±0.48	78.46±2.23	22.72±1.11	79.29±3.01	60.53±4.29	75.29±9.02	54.18±21.89	80.96±3.20	59.29±3.65
GraphText	52.13±0.13	43.62±1.07	72.69±1.90	16.72±1.84	83.69±3.45	72.63±3.14	83.52±2.24	76.45±1.18	–	–
LLaGA	52.08±0.10	33.94±0.46	83.55±0.02	42.34±0.17	88.90±0.09	84.16±0.25	81.00±0.07	72.51±0.25	90.31±0.00	81.37±0.06
RGLM	55.19±0.01	41.54±0.04	83.92±0.04	45.41±0.05	90.20±0.01	86.52±0.01	82.54±0.02	74.61±0.06	90.17±0.00	78.10±0.03
GOFA	56.36±1.22	42.91±4.97	84.62±0.53	48.21±3.43	89.18±0.31	84.27±0.76	86.50±0.17	81.80±0.16	87.64±0.42	82.53±0.81
GFT	57.71±0.08	39.46±0.97	84.50±0.20	44.56±1.10	87.79±0.18	81.49±1.25	87.11±0.13	82.25±0.36	92.23±0.12	86.66±0.28
GRAVER	26.58±1.88	22.66±1.29	62.31±2.92	29.33±0.99	65.35±3.17	56.94±2.75	65.98±4.46	62.82±3.04	52.83±3.03	36.43±2.30
OpenGraph	48.23±1.08	39.92±0.31	76.30±0.48	39.50±1.52	82.92±0.02	78.28±0.20	80.60±1.54	76.84±1.48	86.15±0.17	78.84±0.35
GPM	53.62±0.07	35.15±0.15	84.09±0.13	46.58±0.17	90.38±0.05	86.51±0.21	87.08±0.11	82.44±0.20	92.61±0.07	86.80±0.09
GraphToken$_{\text{GNN}}$	55.85±0.21	42.57±0.99	83.32±0.34	39.54±0.15	90.27±0.10	85.04±0.95	86.64±0.02	82.36±0.23	91.71±0.11	83.83±0.81
GraphToken$_{\text{Transf}}$	56.21±0.48	43.16±0.82	84.09±0.51	41.62±2.50	90.36±0.10	85.02±0.45	87.64±0.10	83.28±0.53	91.79±0.88	83.29±0.13
GraphToken$_{\text{LLM}}$	**57.93±0.43**	**45.32±0.67**	**85.06±0.31**	**52.18±0.84**	**90.83±0.12**	**86.71±0.29**	**88.12±0.14**	**83.74±0.36**	**92.78±0.16**	**86.95±0.23**

Table 2: Link prediction. The best result highlighted in **bold**

Methods	Cora		History		Reddit	
	ROC-AUC	AP	ROC-AUC	AP	ROC-AUC	AP
GCN	78.62±0.34	76.91±0.51	79.36±0.48	79.81±0.44	85.74±0.29	86.88±0.27
GAT	77.31±0.42	75.88±0.47	78.14±0.53	78.42±0.49	84.96±0.31	85.73±0.30
OFA	81.19±0.68	78.73±1.15	82.73±0.43	82.96±0.46	87.42±0.34	88.21±0.31
GFT	81.74±0.39	80.26±0.44	83.46±0.35	83.72±0.36	88.31±0.25	89.02±0.24
RGLM	82.48±0.31	**81.72±0.36**	84.52±0.28	84.83±0.31	89.21±0.20	90.06±0.21
GraphToken	**83.26±0.27**	81.61±0.33	**85.31±0.24**	**85.57±0.26**	**90.08±0.18**	**90.91±0.19**

among the GraphToken instantiations. These results indicate that the learned motif vocabulary provides effective structure-semantic representations for downstream node classification across graphs with different structural and textual characteristics.

Performance on Link Prediction. We further evaluate GraphToken$_{\text{LLM}}$ on supervised link prediction across citation, e-commerce, and social graphs. As shown in Table 2, GraphToken outperforms existing baselines in most cases. The gains across graphs with different structural and textual characteristics indicate that the learned motif-level tokens retain information useful for modeling relations between nodes. Together with the node classification results, these findings demonstrate the effectiveness of GraphToken across different graph prediction tasks.

Transferability. Training across multiple data sources can introduce distributional interference and catastrophic forgetting (Liu et al., 2021), making robust transfer challenging. We therefore evaluate GraphToken$_{\text{LLM}}$ under both cross-dataset and cross-domain settings by training on multiple source graphs and testing on held-out targets. As shown in Table 3, GraphToken outperforms the compared methods in most settings. Performance is generally higher when the source collection contains graphs

Table 3: Node classification results under cross-dataset and cross-domain settings.

Source	S1				S2		S3		S4	
Model / Target	Cora		Reddit		Cora		History		History	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
OpenGraph	82.36±1.18	80.29±1.36	44.19±0.71	41.82±1.51	5.62±0.57	4.17±0.49	77.54±1.12	27.83±2.61	58.91±1.68	9.67±1.32
LLaGA	84.52±0.61	82.84±0.78	47.28±0.53	44.76±1.38	6.74±0.38	5.26±0.31	80.63±0.72	32.41±2.86	61.74±1.47	11.83±1.24
RGLM	86.91±0.33	85.74±0.41	50.87±0.36	48.93±0.92	8.63±0.29	7.36±0.24	82.96±0.48	37.42±2.17	66.28±0.96	**15.61±0.88**
GraphToken	**88.38±0.42**	**87.59±0.51**	**52.15±0.28**	**50.55±1.17**	**9.41±0.26**	**8.10±0.04**	**84.07±0.35**	**38.87±4.17**	**67.36±1.31**	15.37±1.09

Sources: **S1**: Cora, PubMed, arXiv, arXiv23; **S2**: PubMed, arXiv, arXiv23; **S3**: Children, History, Photo; **S4**: Children, Photo.

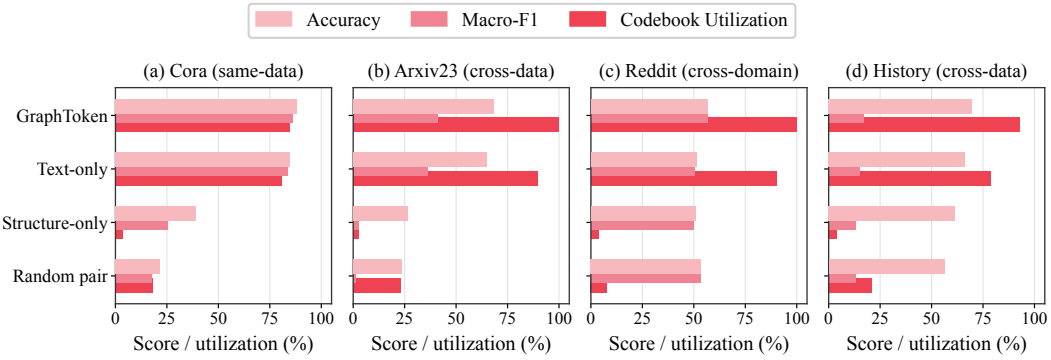


Figure 3: Token Ablation Analysis.

from the target domain, while excluding target-domain data leads to a clear degradation, reflecting the greater difficulty of cross-domain generalization. These results indicate that the learned vocabulary captures reusable structure–text patterns beyond individual source datasets.

5.3 Further Model and Transferability Analysis

Ablation Analysis. To assess the role of structure–text interaction in GraphToken, we conduct cross-dataset and cross-domain ablations across citation, social, and e-commerce graphs, with detailed settings provided in Appendix A.4.1. We compare the full model with *text-only*, which retains semantic information alone, *structure-only*, which relies solely on topology, and *random pairing*, which disrupts structure–text correspondence during tokenizer training. As shown in Figure 3, the full model achieves the most consistent balance between predictive performance and vocabulary utilization. Random pairing markedly degrades both, while structure-only exhibits severe vocabulary collapse and text-only becomes less robust under larger domain shifts. These results highlight aligned structure–text interaction as a key factor in learning a diverse and transferable graph-token vocabulary.

GraphToken Transferability Analysis. To assess the transferability of patterns captured by GraphToken, we evaluate zero-shot vocabulary transfer from Cora and PubMed to Reddit on node classification, with detailed experimental settings provided in Appendix A.4.2. Figure 4 (a) shows that removing the most transferable tokens causes a substantially larger performance drop than removing source-specific or random tokens, indicating that cross-domain predictive utility is concentrated in a small subset of the vocabulary. Figure 4 (b) further attributes this effect primarily to reduced recall on the minority Popular class. These results demonstrate that GraphToken identifies compact, reusable structural units that remain informative under substantial domain shift. Representative top-ranked transferable

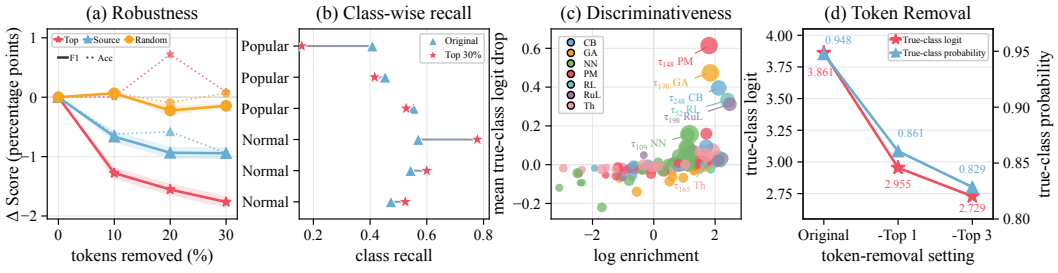


Figure 4: Transferability Analysis.

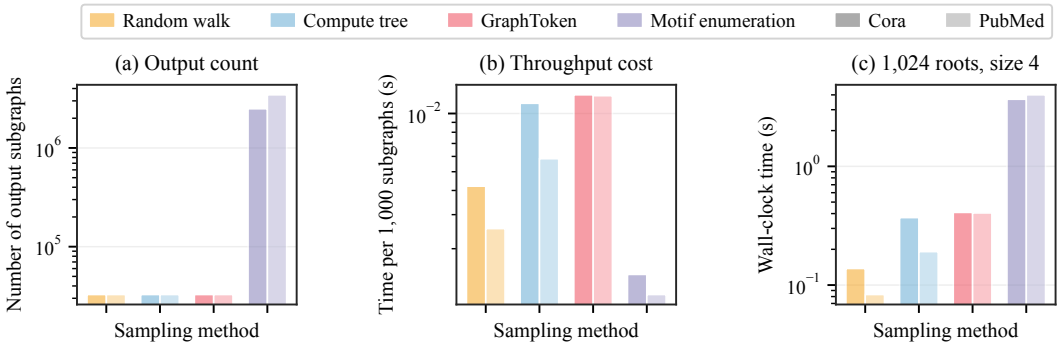


Figure 5: Sampling Efficiency Analysis.

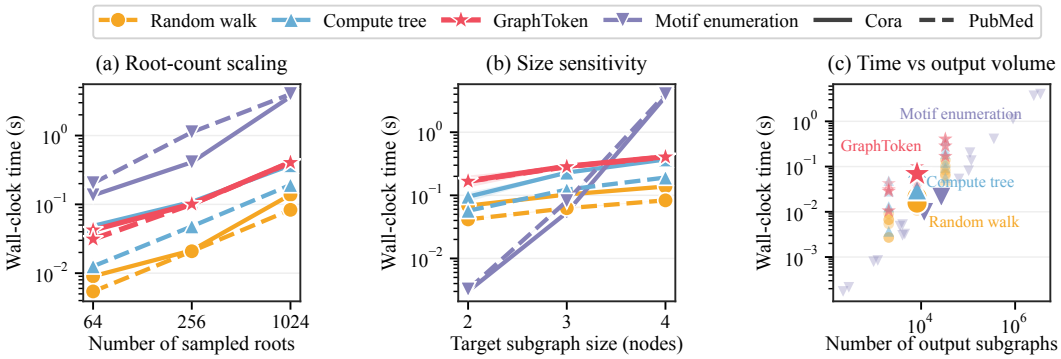


Figure 6: Sampling Scalability Analysis.

graph-tokens are visualized in Appendix A.6.1 Figure 7, while Appendix A.6.2 Figure 8 extends this view with a complete atlas of the nine two- to four-node motif topologies and their occupied GraphToken codes, providing a broader view of the structural coverage of the learned vocabulary.

To examine GraphToken's contribution to class-discriminative predictions, we analyze its learned vocabulary on Cora node classification. Figure 4 (c) relates token specificity to predictive importance, revealing a compact subset of class-selective and decision-relevant graph-tokens amid a broader vocabulary of generic patterns. We further probe their predictive role through node-level counterfactual removal, with the corresponding motif instances visualized in Appendix A.6.3 Figure 9. As shown in Figure 4 (d), progressively removing the most influential tokens consistently reduces the true-class logit and prediction confidence. This alignment between vocabulary-level discriminativeness and instance-level intervention indicates that GraphToken captures interpretable local patterns that provide concrete evidence for downstream decisions.

Sampling Strategy Analysis. To assess the efficiency and scalability of GraphToken sampling, we compare it with random walk (Wang et al., 2025c), computation-tree sampling (Wang et al., 2024), and explicit motif enumeration on Cora and PubMed under varying sampling budgets and subgraph sizes. Figure 5 shows that GraphToken achieves a favorable balance between structural coverage and computational cost, producing richer local subgraphs than lightweight samplers while avoiding the high per-output overhead of explicit enumeration. Figure 6 further shows that this advantage persists as the workload scales, with GraphToken exhibiting controlled runtime growth across increasing numbers of roots and larger target subgraphs. These results establish GraphToken as an efficient and scalable strategy for extracting structurally informative local patterns. Further model complexity analysis is provided in Appendix A.5.

6 Conclusions

In this paper, we introduced **GraphToken**, a motif-based framework for learning a shared discrete vocabulary over text-attributed graphs. By inducing reusable discrete identities from joint structure-text representations of rooted motifs, GraphToken captures recurring local patterns beyond individual graph instances. The resulting vocabulary is architecture-agnostic, providing a common representation layer that can support GNN-, Transformer-, and LLM-based models. This decoupling of vocabulary learning from downstream modeling offers a unified basis for transferable graph learning.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing, including checking grammar, improving readability, and refining the wording of the manuscript. We also used generative AI tools to assist with understanding and debugging code during development. Generative AI was not used to develop the core research idea, formulate the GraphToken methodology, design the main experiments, or generate the core experimental implementation. All AI-assisted edits and debugging suggestions were manually reviewed and verified by the authors. The proposed GraphToken framework, experimental design, implementation of the core method, analysis of the results, and scientific conclusions were developed and validated by the authors. We take responsibility for the final content of this work, including any text or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work focuses on general-purpose representation learning for text-attributed graphs and does not involve human-subject studies or the collection of private or sensitive information. All experiments are conducted on publicly available benchmark datasets. While GraphToken is not designed for high-risk applications, downstream models may inherit biases or limitations present in the underlying graph structure and textual attributes. We therefore encourage careful evaluation when applying the method in sensitive real-world settings.

REPRODUCIBILITY STATEMENT

We provide the details necessary to reproduce our experiments throughout the main paper and appendix. The model architecture, training objectives, and implementation details are described in the methodology section, while dataset statistics, preprocessing procedures, baseline configurations, hyperparameter settings, and additional experimental details are provided in the appendix. We also report results over multiple random seeds to account for experimental variance. An anonymized implementation and configuration files are available at https://anonymous.4open.science/r/GraphToken-15C3.

REFERENCES

Mohammad Sadeq Abolhasani, Yang Ba, Yixuan He, and Rong Pan. Beyond predefined schemas: Trace-kg for context-enriched knowledge graph generation. *arXiv preprint arXiv:2604.03496*, 2026.

Uri Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6): 450–461, 2007.

Austin R. Benson, David F. Gleich, and Jure Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016. doi: 10.1126/science.aad9029.

Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. Llaga: Large language and graph assistant. *arXiv preprint arXiv:2402.08170*, 2024a.

Xuexin Chen, Ruichu Cai, Yuan Fang, Min Wu, Zijian Li, and Zhifeng Hao. Motif graph neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 35(10):14833–14847, 2024b. doi: 10.1109/TNNLS.2023.3281716.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 6022–6034, 2020.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. In *International Conference on Learning Representations*, volume 2024, pp. 31598–31619, 2024.

Zeyuan Guo, Enmao Diao, Cheng Yang, and Chuan Shi. Graph tokenization for bridging graphs and transformers. *arXiv preprint arXiv:2603.11099*, 2026.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

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. *arXiv preprint arXiv:2305.19523*, 2023.

Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 448–459, 2025.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022. doi: 10.1145/3534678.3539321.

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.

Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. Can gnn be good adapter for llms? In *Proceedings of the ACM web conference 2024*, pp. 893–904, 2024.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Lecheng Kong, Jiarui Feng, Hao Liu, Chengsong Huang, Jiaxin Huang, Yixin Chen, and Muhan Zhang. Gofa: A generative one-for-all model for joint graph language modeling. In *International Conference on Learning Representations*, volume 2025, pp. 40792–40823, 2025.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, 2018. doi: 10.18653/v1/D18-2012.

Cheng-Te Li, Yu-Che Tsai, Chih-Yao Chen, and Jay Chiehen Liao. Graph neural networks for tabular data learning: A survey with taxonomy and directions. *ACM Computing Surveys*, 58(1):1–51, 2025.

Ziming Li, Xiaoming Wu, Zehong Wang, Jiazheng Li, Yijun Tian, Jinhe Bi, Yunpu Ma, Yanfang Ye, and Chuxu Zhang. Graph is a substrate across data modalities. *arXiv preprint arXiv:2601.22384*, 2026.

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. *arXiv preprint arXiv:2310.00149*, 2023a.

Huihui Liu, Yiding Yang, and Xinchao Wang. Overcoming catastrophic forgetting in graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 8653–8661, 2021.

Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. Towards graph foundation models: A survey and beyond. *arXiv preprint arXiv:2310.11829*, 2023b.

Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. Graph foundation models: Concepts, opportunities and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002. doi: 10.1126/science.298.5594.824.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1150–1160, 2020. doi: 10.1145/3394486.3403168.

Aravind Sankar, Xinyang Zhang, and Kevin Chen-Chuan Chang. Motif-based convolutional neural network on graphs. *arXiv preprint arXiv:1711.05697*, 2017.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1715–1725, 2016. doi: 10.18653/v1/P16-1162.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.

Chuan Shi, Junze Chen, Jiawei Liu, and Cheng Yang. Graph foundation model. *Frontiers of Computer Science*, 18(6):186355, 2024. doi: 10.1007/s11704-024-40046-0.

Li Sun, Zhenhao Huang, Suyang Zhou, Qiqi Wan, Hao Peng, and Philip S. Yu. RiemannGFM: Learning a graph foundation model from riemannian geometry. In *Proceedings of the ACM on Web Conference 2025*, pp. 1154–1165, 2025. doi: 10.1145/3696410.3714952.

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*, pp. 491–500, 2024.

Vicuna Team. Vicuna: An open-source chatbot impressing gpt-4 with 90% chatgpt quality. *Vicuna: An open-source chatbot impressing gpt-4 with*, 90, 2023.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, and Zhao Kang. Multi-domain graph foundation models: Robust knowledge transfer via topology alignment. *arXiv preprint arXiv:2502.02017*, 2025a.

Zehong Wang, Zheyuan Zhang, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. Gft: Graph foundation model with transferable tree vocabulary. *Advances in neural information processing systems*, 37:107403–107443, 2024.

Zehong Wang, Zheyuan Liu, Tianyi Ma, Jiazheng Li, Zheyuan Zhang, Xingbo Fu, Yiyang Li, Zhengqing Yuan, Wei Song, Yijun Ma, Qingkai Zeng, Xiusi Chen, Jianan Zhao, Jundong Li, Meng Jiang, Pietro Lio, Nitesh V. Chawla, Chuxu Zhang, and Yanfang Ye. Graph foundation models: A comprehensive survey. *arXiv preprint arXiv:2505.15116*, 2025b.

Zehong Wang, Zheyuan Zhang, Tianyi Ma, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. Beyond message passing: Neural graph pattern machine. *arXiv preprint arXiv:2501.18739*, 2025c.

T Wolf. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Lianghao Xia, Ben Kao, and Chao Huang. Opengraph: Towards open graph foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2365–2379, 2024.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

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.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Shuo Yu, Yingbo Wang, Ruolin Li, Guchun Liu, Yanming Shen, Shaoxiong Ji, Bowen Li, Fengling Han, Xiuzhen Zhang, and Feng Xia. Graph2text or graph2token: A perspective of large language models for graph learning. *ACM Transactions on Information Systems*, 44(3):1–49, 2026.

Xingtong Yu, Chang Zhou, Yuan Fang, and Xinming Zhang. Text-free multi-domain graph pre-training: Toward graph foundation models. *arXiv preprint arXiv:2405.13934*, 2024.

Xingtong Yu, Zechuan Gong, Chang Zhou, Yuan Fang, and Hui Zhang. Samgpt: Text-free graph foundation model for multi-domain pre-training and cross-domain adaptation. In *Proceedings of the ACM on Web Conference 2025*, pp. 1142–1153, 2025.

Haonan Yuan, Qingyun Sun, Junhua Shi, Xingcheng Fu, Bryan Hooi, Jianxin Li, and Philip S Yu. Graver: Generative graph vocabularies for robust graph foundation models fine-tuning. *arXiv preprint arXiv:2511.05592*, 2025.

Zhongjian Zhang, Xiao Wang, Mengmei Zhang, Jiarui Tan, and Chuan Shi. Toward graph-tokenizing large language models with reconstructive graph instruction tuning. In *Proceedings of the ACM Web Conference 2026*, pp. 430–441, 2026.

Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4443–4454, 2024.

Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*, 2023.

Xi Zhu, Haochen Xue, Ziwei Zhao, Wujiang Xu, Jingyuan Huang, Minghao Guo, Qifan Wang, Kaixiong Zhou, Imran Razzak, and Yongfeng Zhang. LLM as GNN: Graph vocabulary learning for text-attributed graph foundation models. *arXiv preprint arXiv:2503.03313*, 2025a.

Yun Zhu, Haizhou Shi, Xiaotang Wang, Yongchao Liu, Yaoke Wang, Boci Peng, Chuntao Hong, and Siliang Tang. Graphclip: Enhancing transferability in graph foundation models for text-attributed graphs. In *Proceedings of the ACM on Web Conference 2025*, pp. 2183–2197, 2025b.

A APPENDIX

A.1 DATASETS DETAILS

A.1.1 STATISTICS OF EXPERIMENT DATASETS

To establish the multi-domain training setting, we curated ten benchmark graph datasets spanning three representative domains. These datasets exhibit diverse structural patterns, scales, and textual attributes, enabling a rigorous assessment of model robustness and generalization across graph scenarios. The dataset statistics are in Table 4.

Table 4: Statistics of experiment datasets.

Dataset	# Nodes	# Edges	# Class	Splitting	Domain	Text
Cora	2,708	5,429	7	60/20/20	citation	title, abstract
Pubmed	19,717	44324	3	60/20/20	citation	title, abstract
ogbn-arxiv	169,343	1,157,799	40	54/18/28	citation	title, abstract
tape-arxiv23	46,198	38,863	38	60/20/20	citation	title, abstract
Books-Children	76,875	1,162,522	24	60/20/20	E-commerce	title, description
Books-History	41,551	251,590	12	60/20/20	E-commerce	title, description
Ele-Computers	87,229	628,274	10	60/20/20	E-commerce	product reviews
Ele-Photo	48,362	436,891	12	60/20/20	E-commerce	product reviews
Sports-Fitness	173,055	1,510,067	13	20/10/70	E-commerce	product description
Reddit	33,434	134,236	2	10/10/80	social network	user posts

A.1.2 CITATION DOMAIN.

• **Cora** (McCallum et al., 2000) is a citation graph of computer science publications, where nodes correspond to papers and edges indicate citation links. Each node is equipped with textual attributes derived from the paper's title and abstract, and a class label denoting its research topic.
• **Pubmed** (Sen et al., 2008) is a biomedical citation graph where nodes denote scientific articles and edges encode citation links among papers. Each node includes text features extracted from the title and abstract, along with a label indicating its subject category.
• **ogbn-arxiv** (Hu et al., 2020) is a paper citation graph derived from the arXiv repository, where nodes correspond to individual articles and edges denote citation relations between them.
• **tape-arxiv23** (He et al., 2023) dataset is a citation graph that represents the citation network between all computer science arXiv papers published in 2023 or later. Similar to ogbn-arxiv, each node is an arXiv paper, and each edge indicates that one paper cites another. Each node is annotated with text-based features from the title and abstract, as well as a label indicating the paper's research topic.

A.1.3 E-COMMERCE DOMAIN.

• **Books-Children/History** (Yan et al., 2023) are derived from the Amazon-Books collection by filtering items under the second-tier categories of Children and History. The resulting graphs use books as nodes and connect pairs that are often purchased or viewed together. Each node is annotated with the corresponding three-level book category, while its textual representation is formed from the title and description.
• **Ele-Computers/Photo** (Yan et al., 2023) are two subgraphs extracted from Amazon-Electronics, built from the 2018 Amazon Computer and Amazon Photo datasets. They are formed by filtering products in the Computers and Photo second-level categories, respectively. Nodes represent individual electronic items, and edges capture frequent co-purchase or co-view relationships. Each product carries a three-tier category label, while its textual attribute is derived from customer reviews, using the most-voted review when available; otherwise, a randomly chosen one.
• **Sports-Fitness** (Yan et al., 2023) is a subset of Amazon-Sports formed by selecting products under the second-level category of Fitness. In this graph, each node corresponds to a fitness-related item, and edges indicate that two items are often purchased or viewed together. The target label is the

item's three-level category.

A.1.4 SOCIAL DOMAIN.

• **Reddit** (Huang et al., 2024) is a social interaction graph in which nodes represent users, and edges capture reciprocal reply relationships. Node text features are constructed from up to three of each user's most recent subreddit posts, and each node is assigned a binary label indicating whether the user is classified as popular or normal.

A.2 BASELINE DETAILS

We benchmark GraphToken against 15 state-of-the-art methods spanning four major categories: traditional GNNs, pretrained graph models, graph-language models, and token/pattern-based models. This comparison evaluates GraphToken against representative paradigms of graph representation and foundation modeling.

The traditional GNN models include:

- **GCN** (Kipf & Welling, 2016) learns node representations by propagating and aggregating information from local neighborhoods. It is computationally efficient and has been widely adopted for semi-supervised node classification.
- **GAT** (Veličković et al., 2017) models neighborhood influence through attention-based aggregation, assigning different importance to different neighbors. It is effective for node classification tasks that benefit from adaptive weighting.
- **GraphSAGE** (Hamilton et al., 2017) constructs node embeddings by sampling and aggregating information from local neighborhoods. Its inductive framework enables scalable representation learning on large graphs and unseen nodes.

The pretrained graph models include:

- **GraphMAE** (Hou et al., 2022) introduces a masked graph autoencoder for generative self-supervised graph learning. It reconstructs masked node features rather than graph structures and employs a scaled cosine error to improve training robustness, achieving strong performance across various graph learning tasks.
- **OFA** (Liu et al., 2023a) develops a unified graph learning framework that uses textattributed graphs to align graph data from different domains in a shared representation space. It further introduces nodes-of-interest to unify node-, link-, and graph-level tasks and employs graph prompting for in-context learning, enabling a single model to generalize across diverse domains and tasks.
- **UniGraph** (He et al., 2025) introduces a graph foundation model that unifies diverse text-attributed graphs through textual node representations. It employs a cascaded LM-GNN architecture with masked graph modeling for large-scale self-supervised pretraining and graph instruction tuning for zero-shot prediction, enabling effective transfer to unseen graphs and tasks across different domains.
- **GraphCLIP** (Zhu et al., 2025b) introduces a self-supervised graph foundation model for cross-domain zero- and few-shot learning on text-attributed graphs. It constructs large-scale graph-summary pairs with LLMs and employs contrastive graph-summary pretraining with invariant learning to enhance cross-domain transferability, while introducing graph prompt tuning for efficient few-shot adaptation and achieving strong performance across diverse downstream tasks.

The graph-language models include:

- **GraphText** (Zhao et al., 2023) introduces a framework that translates graphs into natural language through a graph-syntax tree, enabling large language models to perform graph reasoning as text generation. By bridging structured graph data and text space, it supports in-context reasoning and interactive graph understanding.

- **LLaGA** (Chen et al., 2024a) adapts large language models to graph-structured data by converting nodes into structure-aware sequences and projecting them into the token embedding space.
- **RGLM** (Zhang et al., 2026) employs a reconstructive graph instruction tuning pipeline that enforces graph supervision by reconstructing graph information from the LLM's token outputs.
- **GOFA** (Kong et al., 2025) introduces a generative graph language model that interleaves GNN layers with a frozen LLM to jointly model graph structure and semantics. Through graph-specific pretraining and instruction tuning, it achieves strong zero-shot performance on unseen graphs and tasks.

The token/pattern models include:

- **GFT** (Wang et al., 2024) builds a graph foundation model by treating computation trees derived from message passing as transferable graph tokens. This shared tree vocabulary enables cross-task and cross-domain transfer while reducing negative transfer and improving generalization across diverse graph learning settings.
- **GRAVER** (Yuan et al., 2025) builds graph vocabularies using graphon-based generative experts under a universal task template and augments support samples with in-context vocabularies to enable prompt-based fine-tuning.
- **OpenGraph** (Xia et al., 2024) introduces a graph foundation model for zero-shot generalization across diverse graphs. It combines LLM-based data augmentation, a unified graph tokenizer, and a scalable graph Transformer to capture global topology and adapt to unseen graph properties.
- **GPM** (Wang et al., 2025c) bypasses message passing by extracting and encoding task-relevant substructures as graph patterns, improving expressivity, long-range dependency modeling, scalability, interpretability, and out-of-distribution generalization across diverse graph learning tasks.

A.3 Computing Environment and Resources

The experiments are conducted in a computing environment with the following specifications:

- **OS**: Ubuntu 20.04.4 LTS.
- **CPU**: Intel(R) Xeon(R) Platinum 8352V CPU @ 2.10 GHz with 503 GiB of memory.
- **GPU**: NVIDIA A800 80 GB PCIe.
- **Software**: CUDA 12.2, Python 3.10.13, PyTorch 2.4.1, and PyTorch Geometric 2.6.1.

A.4 Implementation Details

GraphToken constructs discrete vocabulary units from sampled rooted motifs rather than explicitly enumerating structure-text configurations. We search the learning rate over $\{1, 2, 3, 5\} \times \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. For the LLM-based variant, we use Vicuna-7B-v1.5-16K (Team, 2023) and implement the language-model components with Hugging Face Transformers (Wolf, 2019). The text encoder is kept frozen throughout GraphToken training, such that the learned vocabulary is driven by the motif encoder and discrete vocabulary induction rather than by task-specific adaptation of the language model. The resulting text feature is shared by all sampled motifs containing the corresponding node. For all baseline methods, we use the official implementations whenever available and follow their default configurations unless otherwise specified.

GraphToken is evaluated on node classification and link prediction across the ten datasets described in Appendix A.1.1. For the reported GraphToken results, we average performance over 5 independent runs. Node classification is evaluated using Accuracy and Macro-F1. Link prediction uses ROC-AUC and AP under a fixed 8:1:1 train/validation/test split, with an equal number of negative edges sampled from unconnected node pairs and kept disjoint across splits. For transfer experiments, the rooted-motif encoder and discrete vocabulary are learned only from the source datasets and then frozen on the target dataset. A target-specific prediction head is trained using the target training split, while the tokenizer remains unchanged. Evaluation is performed on the target test split.

A.4.1 Ablation Setting.

We evaluate four tokenization variants on one within-dataset task and three cross-dataset transfer settings spanning citation, social, and e-commerce graphs, including Cora node classification, Cora+PubMed+ogbn-arxiv $\rightarrow$ arXiv23, Cora+PubMed $\rightarrow$ Reddit, and Children+Photo+Fitness $\rightarrow$ History. For Cora, the tokenizer, codebook, and classification head are trained and evaluated within the same dataset.

The ablations disentangle structure, text, and their correspondence. Full jointly models the aligned topology and node text of each rooted motif. Text-only retains semantic inputs while removing structural cues, whereas structure-only preserves topology-derived information with content-independent text. Random pairing preserves both modalities but disrupts their correspondence during tokenizer training by pairing motif structures with text from other motifs of matched size. We report Accuracy and Macro-F1 together with codebook utilization, allowing downstream performance to be assessed alongside the diversity and effective coverage of the learned vocabulary.

A.4.2 Token Importance and Transferability Metrics

We distinguish label-dependent interpretability metrics from label-free transferability scores. Predictive importance and class specificity are used only for post-hoc analysis, whereas transferable-token rankings are computed without target labels.

For node v, the importance of token τ_k is measured by the decrease in the ground-truth logit after removing all motif-token embeddings $\mathbf{e}_{c(v,i)}$ assigned to τ_k,

$$I_{v,k} = f_{y_v}(v) - f_{y_v}(v \setminus \tau_k).$$

Class-level importance averages $I_{v,k}$ over nodes of class c containing τ_k. Class specificity is measured by log enrichment,

$$E_{c,k} = \log \frac{\widehat{P}(c \mid \tau_k)}{\widehat{P}(c)}.$$

Positive $E_{c,k}$ indicates a preferential association between token τ_k and class c. Both quantities are used exclusively for post-hoc interpretation.

For cross-domain analysis, transferable tokens are ranked using only unlabeled source and target representations. The score combines source–target prevalence overlap O_k, target quantization distance $D_{k,t}$, source–target motif discrepancy Δ_k, and target assignment confidence $C_{k,t}$,

$$T_k = \frac{1}{4}\left(O_k + \frac{1}{1+D_{k,t}} + \frac{1}{1+\Delta_k} + C_{k,t}\right).$$

Higher T_k reflects stronger cross-domain consistency in token occurrence and representation. Source-specific tokens are conversely characterized by high source prevalence, large source-target discrepancy, and low prevalence overlap. Neither transferability nor source-specificity uses target labels or target-side prediction scores. Target labels are accessed only after ranking for downstream evaluation, preventing target-label leakage.

A.4.3 Rooted Motif Sampling Details

We construct rooted motif instances through topology-driven stochastic expansion from each target node. For a target node v, the i-th sampling trial is initialized as $\mathcal{V}_{v,i}^{(1)} = \{v\}$. At expansion step t, we define the boundary edges connecting the current motif to unseen nodes as $\mathcal{B}\left(\mathcal{V}_{v,i}^{(t)}\right) = \left\{(u,w) \in \mathcal{E} \;\middle|\; u \in \mathcal{V}_{v,i}^{(t)},\ w \notin \mathcal{V}_{v,i}^{(t)}\right\}$. We uniformly sample a boundary edge, $(u,w) \sim \mathrm{Uniform}\left(\mathcal{B}\left(\mathcal{V}_{v,i}^{(t)}\right)\right)$, and expand the motif by adding the newly reached node $\mathcal{V}_{v,i}^{(t+1)} = \mathcal{V}_{v,i}^{(t)} \cup \{w\}$. Expansion continues until the prescribed motif size is reached or no valid boundary edge remains. The corresponding edge set is inherited from the original graph, yielding the rooted motif

$$m_{v,i} = (\mathcal{V}_{v,i}, \mathcal{E}_{v,i}, \mathcal{T}_{v,i}, v), \tag{8}$$

where $\mathcal{T}_{v,i} = \{t_u \mid u \in \mathcal{V}_{v,i}\}$ contains the textual attributes of the sampled nodes. We consider motifs containing two to four nodes and perform $M = 32$ independent sampling trials per target

node, with nominal budgets $(M_2, M_3, M_4) = (4, 8, 20)$. A trial may terminate early when the local neighborhood cannot support further expansion. Nodes are not repeated within a motif, whereas identical rooted motifs may be sampled across different trials. Sampling depends only on graph topology and is independent of node labels or predefined motif categories, allowing paths, stars, triangles, cycles, and other local configurations to emerge naturally from the neighborhood of the root.

A.4.4 ROOTED STRUCTURAL TYPES.

Each sampled motif $m_{v,i}$ is assigned a categorical rooted motif type $\rho_{v,i} \in \{0, \ldots, 15\}$. The type ID is determined deterministically from the motif size, internal connectivity, and the structural position of the designated root under root-preserving canonical topology. Consequently, motifs that are isomorphic as rooted graphs share the same type regardless of their original node identities, whereas motifs with the same unrooted topology may receive different type IDs when the root occupies structurally distinct positions. We use a learnable embedding table with 16 entries to encode these rooted motif types.

A.5 COMPLEXITY ANALYSIS

Let $N = |\mathcal{V}|$ and $E = |\mathcal{E}|$ denote the numbers of nodes and edges, respectively. Conventional GNNs, such as GCN (Kipf & Welling, 2016), GraphSAGE (Hamilton et al., 2017), and GAT (Veličković et al., 2017), have a typical cost of $\mathcal{O}(L_g(Ed + Nd^2))$, while GraphMAE (Hou et al., 2022) introduces additional reconstruction with the same asymptotic graph-size dependence. UniGraph (He et al., 2025) further incurs node-text encoding, giving approximately $\mathcal{O}(NC_{\text{text}} + L_g(Ed + Nd^2))$. GOFA (Kong et al., 2025) combines graph propagation with node-wise language modeling, whose attention cost scales with the total number of node-level text tokens rather than global graph-wide attention. GraphCLIP (Zhu et al., 2025b) additionally performs graph–text contrastive alignment, while OpenGraph (Xia et al., 2024) employs scalable anchor-based attention to avoid quadratic attention over all graph tokens. LLM-based methods such as GraphText (Zhao et al., 2023) and LLaGA (Chen et al., 2024a) are primarily governed by Transformer inference, with quadratic dependence on the constructed sequence length.

For GraphToken, let M be the number of sampled motifs per target node, s the maximum motif size, K the codebook size, and L_m the number of motif-encoder layers. Its tokenization complexity is $\mathcal{O}\left(NM\left[L_m(s^2d + sd^2) + Kd\right]\right)$, excluding one-time node-text encoding. Since M, s, and K are fixed independently of graph size, GraphToken scales linearly with the number of target nodes. Moreover, bounded motif sampling avoids exhaustive motif enumeration, whose worst-case candidate space can grow as $\mathcal{O}(N\Delta^{s-1})$, where Δ is the maximum node degree.

A.6 CASE STUDY

A.6.1 TRANSFERABLE GRAPHTOKEN PROTOTYPES.

To interpret the structural patterns encoded by transferable GraphTokens, we visualize representative top-ranked tokens across Cora, PubMed, and Reddit in Figure 7. The selected tokens span several characteristic local structures, including an edge, a three-leaf star, a three-node path, and a triangle, providing an intuitive view of the structural regularities preserved by the shared vocabulary.

Each row represents a shared GraphToken, and the three columns show its nearest observed motif instance in Cora, PubMed, and Reddit. For each domain, we select the assigned instance with the smallest embedding-to-prototype distance. Orange marks the root node and blue-grey the remaining nodes, while node labels denote dataset-specific IDs. The model uses the raw textual attributes of the nodes, whereas representative root-node keywords are shown below each motif for compact visualization. For each prototype, d_1 measures the distance between the selected motif embedding and its assigned prototype, while $\Delta d = d_2 - d_1$ denotes the margin to the second-nearest prototype. Smaller d_1 indicates greater representativeness, whereas a larger positive margin reflects a more confident and less ambiguous assignment. These quantities characterize prototype fidelity and quantization confidence rather than downstream predictive importance.

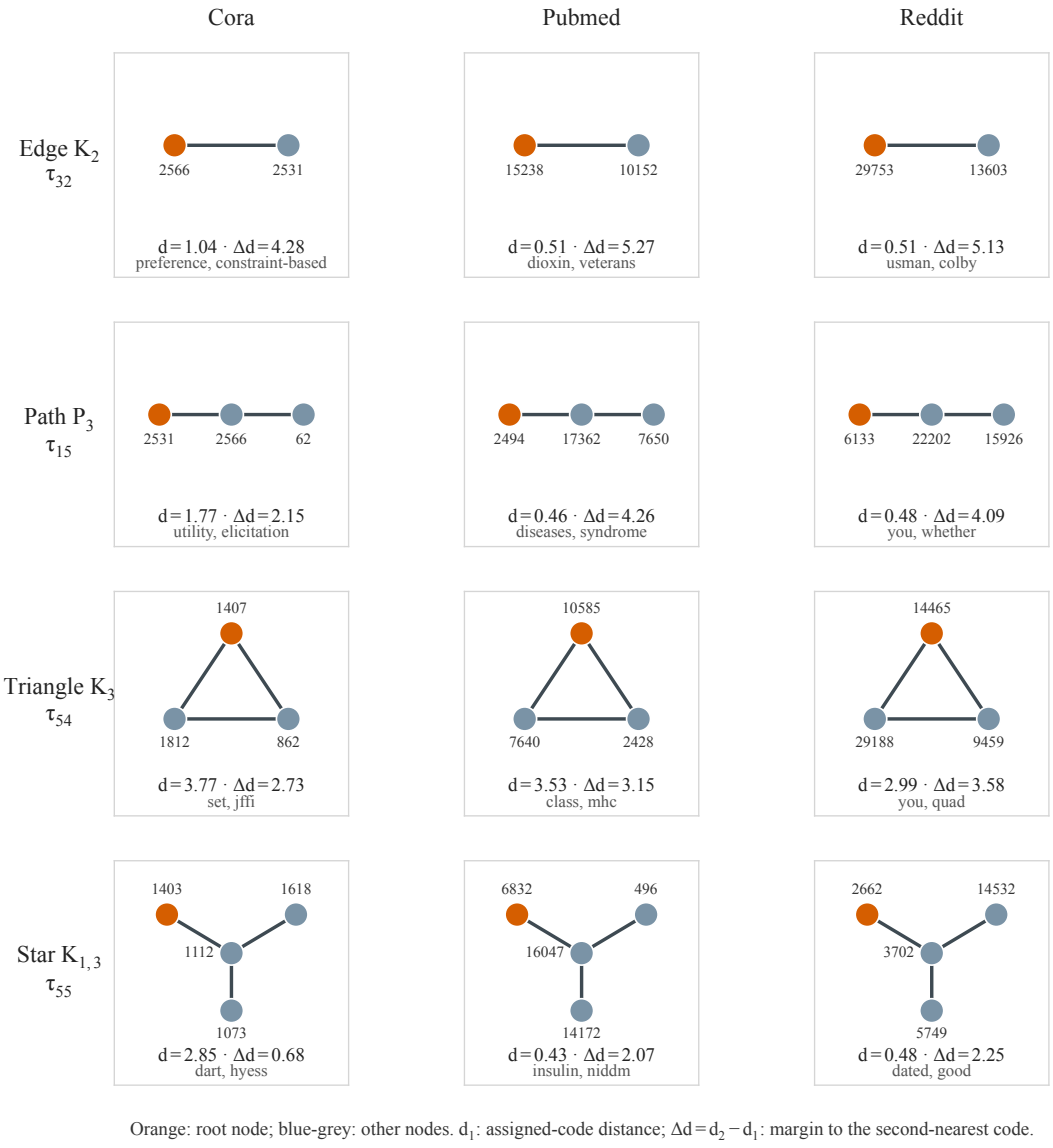
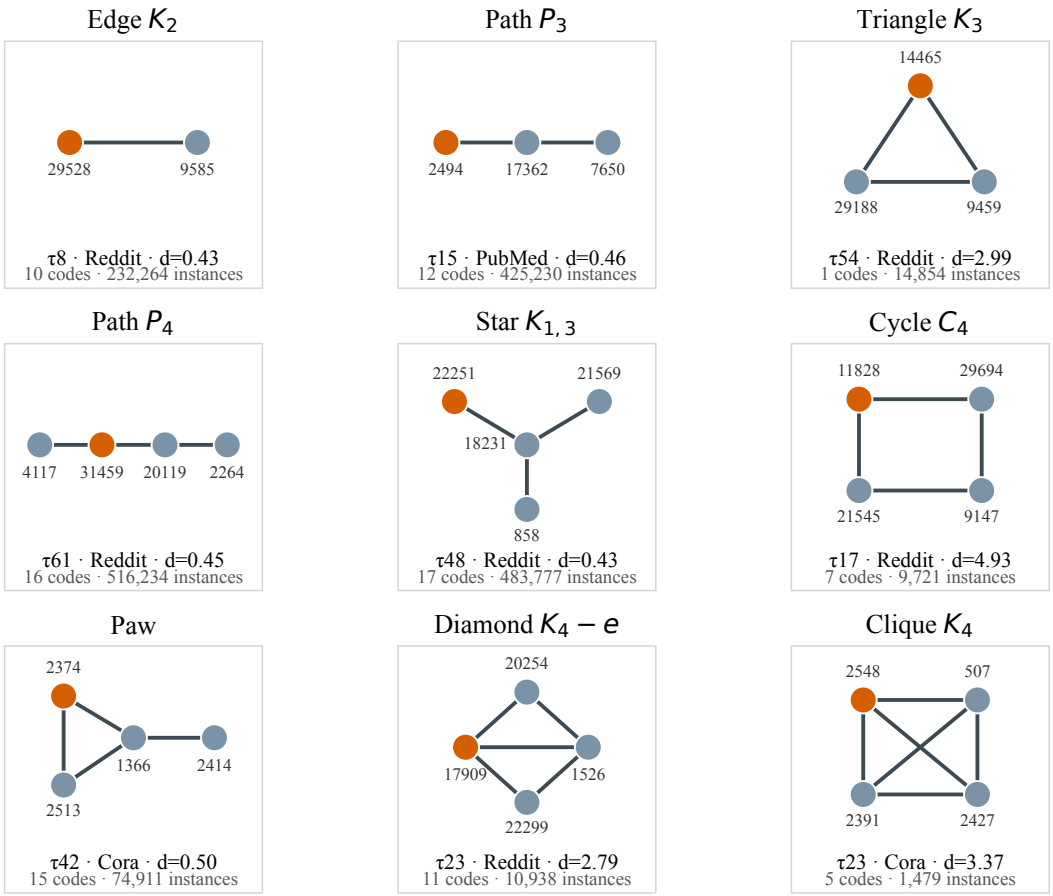


Figure 7: Transferable GraphToken Prototypes.

Despite the substantial shift from the citation networks Cora and PubMed to the Reddit social network, each shared code consistently recovers the same motif topology across domains, while node identities and textual semantics vary. This structural consistency reveals that transferable GraphTokens capture reusable local primitives rather than source-specific patterns. The counterfactual analysis complements this view by establishing their predictive relevance, whereas the prototype visualization exposes the coherent structural regularities underlying that transfer.

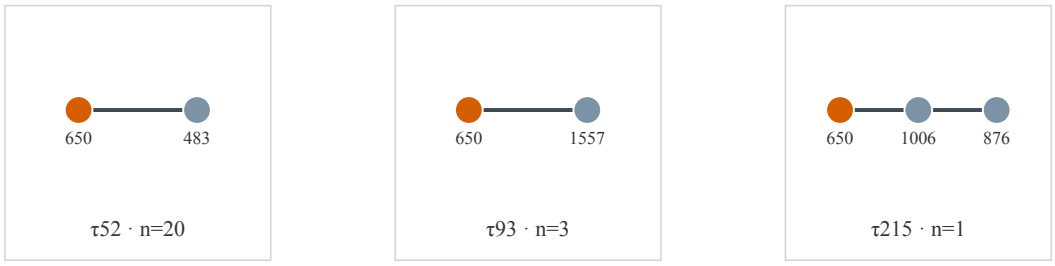
A.6.2 GraphToken vocabulary coverage.

To characterize the structural coverage of the learned vocabulary, we visualize all nine connected motif topologies with two to four nodes in Figure 8, spanning both common sparse patterns and less frequent densely connected structures. GraphToken assigns vocabulary entries across the full topology set, indicating that the learned vocabulary is not dominated by a narrow subset of frequent motifs. Moreover, instances with the same topology can occupy different GraphToken identities, showing that token assignment is governed by the learned motif representation rather than topology alone. Each panel displays the observed instance closest to its assigned prototype, where τ_k denotes the GraphToken and d the embedding-to-prototype distance. Smaller d indicates a more representative instance, with d serving as a diagnostic of prototype compactness.



Orange: root node; blue-grey: other nodes. Each panel shows the instance closest to its codebook center.

Figure 8: GraphToken vocabulary coverage.



Orange: root node; blue-grey: other nodes. n: number of motif instances of the token for node 650.

Figure 9: GraphToken Prototypes for Cora.

A.6.3 GraphTokens for a Cora Node.

To expose the concrete structural patterns underlying the counterfactual effects observed in the main text, we further visualize the three highest-impact GraphTokens for Cora test node 650 in Figure 9, whose ground-truth class is Reinforcement Learning. Each panel displays one representative rooted motif instance assigned to the corresponding codebook entry. The orange node denotes the root node being classified, whereas blue-grey nodes denote the other nodes participating in the motif; the numbers beside the nodes are their original Cora node IDs. The symbol (τ_k) identifies the (k)-th GraphToken, and (n) denotes the total number of motif instances associated with that token in the representation of node 650, rather than the number of nodes displayed in the panel.

These examples reveal that the prediction for node 650 is supported by a heterogeneous but structured collection of local motifs: a dominant and repeatedly occurring token (τ_{52}), a less frequent edge token (τ_{93}), and a higher-order path token (τ_{215}). Their visualization provides a structural interpretation of the removal experiment and illustrates how GraphToken compresses multiple motif instances into reusable discrete codes while preserving their relevance to the downstream prediction.