GILT: An LLM-Free, Tuning-Free Graph Foundational Model for In-Context Learning

Weishuo Ma [1] **Yanbo Wang** [2] **Xiyuan Wang** [2] **Lei Zou** [1] **Muhan Zhang** [2]

Abstract

Graph Neural Networks (GNNs) are powerful tools for processing relational data but often struggle to generalize to unseen graphs, giving rise to the development of Graph Foundational Models (GFMs). However, current GFMs are challenged by the extreme heterogeneity of graph data, where each graph can possess a unique feature space, label set, and topology. To address this, two main paradigms have emerged. The first leverages Large Language Models (LLMs), but is fundamentally text-dependent, thus struggles to handle the numerical features in vast graphs. The second pre-trains a structure-based model, but the adaptation to new tasks typically requires a costly, per-graph tuning stage, creating a critical efficiency bottleneck. In this work, we move beyond these limitations and introduce **G**raph **I**n-context **L**earning **T**ransformer (GILT), a framework built on an LLM-free and tuning-free architecture. GILT introduces a novel token-based framework for in-context learning (ICL) on graphs, reframing classification tasks spanning node, edge and graph levels in a unified framework. This mechanism is the key to handling heterogeneity, as it is designed to operate on generic numerical features. Further, its ability to understand class semantics dynamically from the context enables tuning-free adaptation. Comprehensive experiments show that GILT achieves stronger few-shot performance with significantly less time than LLM-based or tuning-based baselines, validating the effectiveness of our approach. Our code is available at: https://github.com/yiming421/inductnode/.

[1]Wangxuan Institute of Computer Technology, Peking University [2]Institute for Artificial Intelligence, Peking University. Correspondence to: Muhan Zhang <muhan@pku.edu.cn>.

Accepted as an oral presentation at the GFM @ ICML 2026 Workshop.

1. Introduction

Graph Neural Networks (GNNs) have emerged as the standard for processing graph data, achieving state-of-the-art performance on a wide range of single-graph tasks (Kipf & Welling, 2017; 2016; Velickovic et al., 2018; Xu et al., 2019). However, their fundamental limitation is a lack of generalization: a GNN trained on one graph often fails to transfer to an unseen graph with different features or topology (Hu et al., 2020b). In parallel, the artificial intelligence area has been reshaped by the success of foundation models that exhibit remarkable transferability in domains like language (Brown et al., 2020) and vision (Radford et al., 2021). This confluence of GNNs' limitations and the power of the foundation model paradigm has spurred intense interest in a new frontier: the Graph Foundational Model (GFM).

However, the extreme heterogeneity in graph data presents a fundamental obstacle to realizing this vision. Unlike text or images which benefit from a universal vocabulary, graphs lack this common foundation. Each graph can possess its own arbitrary feature space, with varying dimensions and semantics; a distinct target space with its own unique structure like discrete classes or continuous values; and a vastly different topological structure (Mao et al., 2024). Consequently, the parameters of a conventional GNN architecture are fundamentally tied to the specific feature and output formats of its training data, making the model inherently non-transferable (Hu et al., 2020b). This core challenge of bridging graph heterogeneity has driven the development of the main GFM paradigms to date (Liu et al., 2025).

The first primary paradigm tackles the heterogeneity problem by leveraging Large Language Models (LLMs) to create a unified semantic space (Li et al., 2024a; Fan et al., 2024; Ren et al., 2024). The core strategy involves applying an LLM to interpret the textual information associated with a graph's nodes and edges, mapping diverse features and labels into a shared space. Although effective for text-rich graphs like citation networks, this approach introduces a dependency on textual data (Zhao et al., 2025; 2024a; Sun et al., 2025a). Consequently, it is unsuitable for graphs with numerical, categorical, or purely structural data, as is common in fields like molecular biology (Wu et al., 2017).

The second major paradigm, known as graph prompting, takes a more direct and graph-native solution to heterogeneity by real-time parameter adaptation, typically involving pre-training a GNN encoder on large-scale data and then adapting it to downstream tasks (Sun et al., 2023a; 2025a; Yu et al., 2025a). While this approach avoids the text-dependency issue, it introduces a dependency on tuning. The need to modify model weights for each new graph or task persists (Sun et al., 2023b), creating a significant efficiency bottleneck and diverging from the promise of a truly "out-of-the-box" foundational model (Zhao et al., 2025).

In this work, we move beyond the two barriers by introducing the **Graph In-context Learning Transformer (GILT)**, a framework designed to be both **LLM-free and tuning-free**. Our key innovation is to reframe few-shot graph tasks, spanning node, edge, and graph classification, as a unified token-based in-context learning problem. GILT's architecture first tokenizes a task, converting its structure and features into a set of tokens. These tokens are then processed by a Transformer that learns the task's semantics directly from the prompted examples. This in-context learning mechanism allows GILT to dynamically interpret unseen feature and label spaces at inference time, completely bypassing the need for textual information or parameter updates.

We validate GILT on diverse benchmarks spanning node, link, and graph classification, with results confirming its state-of-the-art few-shot performance. Its LLM-free design allows it to operate directly on text-independent graphs with numerical features, where text-based models are often either inapplicable or require laborious pre-processing to manually create textual descriptions. Concurrently, its tuning-free nature provides a significant efficiency advantage, making it orders of magnitude faster than methods that require per-graph tuning and positioning it as a more scalable solution. Our code is available at: `https://github.com/yiming421/inductnode/`.

Our main contributions in this work are as follows:

- We design and implement GILT, a LLM-free, tuning-free In-Context Learning architecture, reframing few-shot graph problems as a unified token-reasoning task.
- We collect diverse datasets for pretraining and trained one model for multiple tasks through a graph-native tokenization pipeline and a two-stage ICL Transformer.
- We provide comprehensive empirical validation establishing GILT's superiority. Our experiments demonstrate its state-of-the-art few-shot performance in text-free graphs, and its efficiency on faster than both tuning-based adaptation and the heavy inference required by LLMs.

2. Related Work

The development of GFMs can be understood through two lenses: **the core techniques** used in its architecture and **the target learning paradigm**, which defines how a model adapts to new tasks. This section reviews the field along these lines, showing how limitations of current techniques motivate a paradigm shift towards in-context learning.

2.1. Popular Techniques in GFM Architecture

To address key challenges like graph heterogeneity and task adaptation, researchers have developed several powerful techniques that are often used in combination.

LLMs for GFMs A primary research direction for building GFMs involves unifying heterogeneous graph data within the text domain to leverage the capabilities of LLMs. These efforts can be broadly grouped into two strategies. The first uses an **LLM as an enhancer**, processing diverse textual node features into a common representation space before performing structure-aware prediction (Huang et al., 2023; Liu et al., 2024; Chen et al., 2023; Li et al., 2024b; He et al., 2024; Wang et al., 2024c; Plenz & Frank, 2024; Zhu et al., 2025b; Xia et al., 2024; Wang et al., 2025a). For instance, ZeroG (Li et al., 2024b) leverages an LLM to encode textual node attributes into a unified semantic space and then performs neighborhood aggregation for the final prediction. The second strategy employs the **LLM as the predictor**, aiming to leverage its generative capabilities for greater task flexibility (Tang et al., 2024; Chen et al., 2024; He et al., 2025a; Zhang et al., 2024b; Kong et al., 2025; Hu et al., 2024; He et al., 2025b; Wang et al., 2024a;b; Zhang et al., 2024a; Sun et al., 2025b). The GOFA model (Kong et al., 2025) achieves this by interleaving GNN layers into an LLM's architecture, combining message-passing with semantic reasoning before generating prediction. Ultimately, the reliance of both strategies on a textual foundation naturally limits their scope to text-attributed graphs.

Graph Prompting Another dominant technique is graph prompting, which introduces learnable structural components that steer a pre-trained model's reasoning across diverse tasks. Following an initial large-scale pre-training phase, the GNN's weights are frozen. Adaptation is then achieved by introducing and tuning small, learnable prompts that adapt the model's behavior for new tasks (Sun et al., 2022; Liu et al., 2023; Sun et al., 2023a; Fang et al., 2023; Zi et al., 2024; Zhao et al., 2024a; Sun et al., 2025a; Zhao et al., 2024b; Yu et al., 2025a; Zhu et al., 2025a; Lin et al., 2025; Yu et al., 2025b; Chen et al., 2025; Wang et al., 2025b; Yang et al., 2025; Yuan et al., 2025). For instance, the GCOPE (Zhao et al., 2024a) framework extends a prompt-like mechanism to the pre-training stage, using learnable coordinators as virtual nodes to align different graph datasets. RiemannGFM (Sun et al., 2025a) introduces a novel geometric perspective, pre-training a model on a universal "structural vocabulary" of trees and cycles, which is then adapted to new tasks through prompt tuning. While parameter-efficient,

Table 1. Comparison of GILT with representative Graph Foundational Model paradigms.

Method / Paradigm	LLM-Free	Tuning-Free	Multi-Domain Pre-training	Multi-Task	Few-shot
ZeroG	✗	✓	✓	✗	✗
GOFA	✗	✓	✓	✓	✗
GCOPE	✓	✗	✓	✓	✓
RiemannGFM	✓	✗	✓	✓	✓
OFA	✗	✓	✓	✓	✓
GraphAny	✓	✓	✗	✗	✓
GILT (Ours)	✓	✓	✓	✓	✓

these methods' adaptation process still hinges on **gradient-based updates for each new graph**.

2.2. The Paradigm Shift Towards In-Context Learning

The ultimate goal for a GFM is to operate as a ready-to-use system that generalizes to new tasks without re-training. This has motivated a shift towards In-Context Learning, a paradigm where a pre-trained model solves a new task at inference time using only a few prompted examples, without any parameter updates (Brown et al., 2020). While popularized by LLMs, the success of ICL on structured tabular data (Hollmann et al., 2022; 2025), computer vision (Wang et al., 2023), and time-series (Lu et al., 2025) has underscored its potential for more modalities.

Pioneering frameworks have begun to explore this direction. OFA (Liu et al., 2024), for instance, enables in-context learning through constructing a unified prompt graph connecting labeled support examples to their class nodes, allowing a GNN to synthesize this structural context for classification in a single forward pass. More recently, GraphAny (Zhao et al., 2025) achieved tuning-free generalization by using a pre-trained attention module to fuse the outputs of multiple non-parametric, analytical solvers. These works established the viability of tuning-free adaptation on graphs and laid the groundwork for more advanced ICL systems.

Building on these insights, GILT introduces a more general and powerful framework for in-context learning on graphs which uses a Transformer backbone chosen for its proven strength in ICL across language and tabular domains. With the aid of a specialized graph encoding module, the framework is inherently LLM-free to learn from raw numerical features alone. Its deep, pre-trained model is designed to learn complex non-linear patterns, making it a flexible solution for node, link, and graph classification tasks.

3. Method

To overcome the text and tuning barriers inherent in current GFMs, we introduce the GILT framework. The core technical contribution of our work is to reframe the few-shot graph learning task as a problem of reasoning over a set of contextual tokens, allowing us to leverage the power of the Transformer for universal in-context reasoning on graphs.

We formally define a graph as $G = (V, E)$ with node features $X \in \mathbb{R}^{|V| \times d_{in}}$ and an adjacency matrix A. GILT is designed for the **few-shot, in-context learning** setting. Each task is an N-way K-shot problem, where the model is given a **support set** of labeled examples, $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N \times K}$, and must predict labels for an unseen **query set**, $\mathcal{Q} = \{x_j\}_{j=1}^{Q}$. The items x_i can be nodes, edges, or entire graphs. The model must leverage the context from $\mathcal{S}$ to make predictions for $\mathcal{Q}$ without any parameter updates.

3.1. Architecture Overview

As illustrated in Figure 1, GILT is a two-phase pipeline designed to first translate graph tasks into a universal format and then reason over that format to make predictions.

Phase 1: Graph-Native Tokenization (Syntactic Unification). The first phase tackles graph heterogeneity. It converts a raw, few-shot graph task with its unique features and structure into a standardized set of contextual tokens. This creates a unified format that represents the problem.

Phase 2: In-Context Reasoning (Semantic Unification). The second phase is designed to understand the task's meaning from the tokens alone without any tuning. A specialized ICL Transformer processes the token set, learning the task's rules from the provided examples to make a prediction.

3.2. Graph-Native Tokenization

The Graph-Native Tokenization phase performs **syntactic unification**, converts raw graphs and the few-shot task definition into the set-based representation required by the ICL reasoning phase. This section details the components of this phase: the extraction of structural information, and the final task-specific asymmetric token formulation.

Extraction of Structure Information. The tokenization

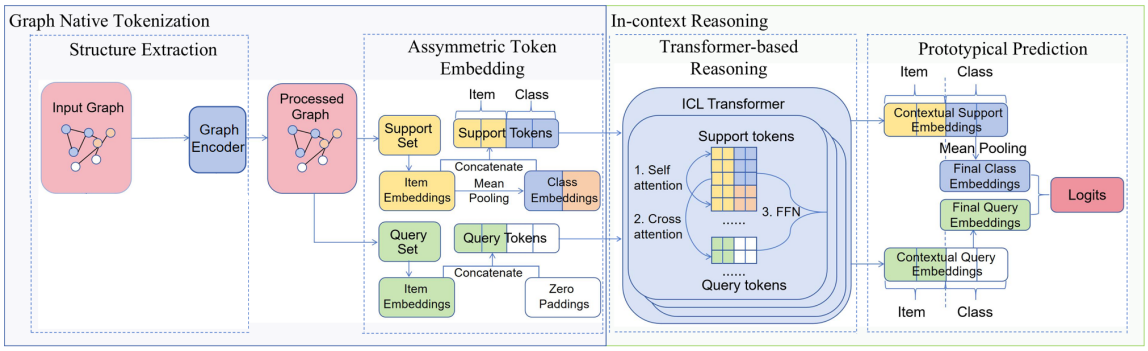

Figure 1. GILT begins with a graph-native tokenization module converting a few-shot task into unified tokens. This module first uses a GNN to generate structure-aware embeddings, which are then combined with class prototypes to form the support and query tokens. The tokens are then passed to ICL Transformer, which features a two-stage attention mechanism for in-context reasoning and a Prototypical Head for the final classification.

pipeline begins with a structural encoder to inject topological context before forming task-specific tokens. The processed raw feature matrix is enriched with topological information by our structural encoder, a deep, multi-layer Graph Convolutional Network (GCN). Critically, we employ a **linear version of the GCN**, omitting the learnable weight matrices and non-linear activations. This design aligns with simplified GCN architectures such as SGC (Wu et al., 2019) and APPNP (Klicpera et al., 2019), which demonstrate that stripping away non-linearities can mitigate overfitting while effectively capturing structural information. The rationale for this choice is rooted in our design principle: a learnable projection at this stage tends to overfit the feature semantics of the pre-training graphs, hindering generalization. By using a simple, parameter-free aggregator, we ensure its role is to strictly extract local structural patterns, deferring all complex semantic reasoning to the more powerful ICL Transformer. We also find that a deeper encoder (4-6 layers) provides a richer, multi-hop context that is highly beneficial for the downstream Transformer. To stabilize the representations as they propagate through the deep encoder, each linear aggregation is followed by an independent LayerNorm with its own learnable affine parameters:

$$H^{(l+1)} = \text{LayerNorm}(\tilde{A}H^{(l)}) \tag{1}$$

where $\tilde{A}$ is the normalized adjacency matrix with self-loops and $H^{(0)} = X'$. The output of the final layer is the node embedding matrix $H \in \mathbb{R}^{|V|\times d}$.

Asymmetric Token Formulation. The next step is to encode the entire N-way K-shot task into a set of fixed-dimension tokens. We derive a single vector representation, denoted h, for each task item (a node's embedding for node tasks, an element-wise product for link tasks, or a pooled vector for graph tasks). These item representations, h_i, are then used to form the final tokens through an asymmetric process. An initial representation for each class p_c is computed using simple **mean pooling** over the support item representations, followed by L2 normalization for stability. This class representation is then paired with each support item's representation to form the final support tokens, while query item representations are paired with zero-padding:

$$\begin{aligned} \mathbf{p}_c &= \frac{\frac{1}{|\mathcal{S}_c|}\sum_{(x_i,y_i)\in\mathcal{S}_c}\mathbf{h}_i}{\left\|\frac{1}{|\mathcal{S}_c|}\sum_{(x_i,y_i)\in\mathcal{S}_c}\mathbf{h}_i\right\|_2}, \\ \mathbf{t}_s &= [\mathbf{h}_i \| \mathbf{p}_{y_i}], \qquad \mathbf{t}_q = [\mathbf{h}_j \| \mathbf{0}]. \end{aligned} \tag{2}$$

This prototypical formulation solves a core challenge. Alternatives like one-hot encoding would result in a *variable token dimension*, while decomposing the task into binary problems would prevent the model from reasoning about *inter-class relationships*. Our approach ensures a consistent token size while enabling the Transformer to reason over all class concepts in a shared context.

3.3. In-context Reasoning

With the task syntactically unified into tokens, the In-Context Prediction phase is designed to perform the more complex challenge of **semantic unification**. This is accomplished by two key components: The ICL Transformer performs the contextual reasoning, learning a task-specific mapping from the prompted examples. This is followed by the Prototypical Head, which provides a dynamic classification mechanism adapting to any N-way task.

Transformer-based Reasoning. The ICL Transformer consists of a stack of L identical layers designed to process the unified set of support and query tokens and produce context-aware embeddings. The design is inspired by the principle of causal attention masking, which has been proven essential for in-context learning on structured tabular data by TabPFN (Hollmann et al., 2025). This principle ensures the query items do not influence the representation of the support set, nor should they influence each other. We implement this via a specialized **two-stage process**.

Stage 1: Context Refinement: The first stage of each layer builds a rich, task-specific context from the support set. To achieve this, a multi-head self-attention mechanism is applied exclusively to the set of support tokens, $T_{\mathcal{S}}$. This allows the support examples to interact and form a coherent representation of the task's semantics. The output is a set of refined support embeddings, $T'_{\mathcal{S}}$:

$$T'_{\mathcal{S}} = \text{SelfAttention}(T_{\mathcal{S}}) \tag{3}$$

Stage 2: Information Gathering: This stage uses the refined context to inform the representation of the query tokens. This is the core in-context learning step where the model applies its understanding to the prediction targets. We use a multi-head cross-attention mechanism where the query tokens $T_{\mathcal{Q}}$ serve as the query, while the refined support embeddings $T'_{\mathcal{S}}$ serve as both the keys and values:

$$T'_{\mathcal{Q}} = \text{CrossAttention}(Q = T_{\mathcal{Q}}, K = T'_{\mathcal{S}}, V = T'_{\mathcal{S}}) \tag{4}$$

As is standard, residual connections and LayerNorm are applied around each sub-module to ensure stable training.

Prototypical Prediction. The final stage of our framework is the Prototypical Head, which performs the final, tuning-free classification. Our architecture is designed to maintain a separation of roles within each token embedding: the "item space" serves as the primary input for reasoning, while the "class space" serves as the dedicated output space. The ICL Transformer learns to project its final, context-aware prediction into this class-space portion of its output embeddings. Therefore, for the final prediction, **we use only the class-space portion of the embeddings from the Transformer**. The final classification then proceeds as follows: First, a prototype vector p_c for each class c is computed by taking the element-wise mean of the class-space portions of all final support embeddings. Each query's final class-space embedding is then classified based on its cosine similarity to each class prototype, and these scores are converted into a probability distribution via a softmax function. This mechanism is non-parametric and allows GILT to adapt to any N-way classification task on the fly.

3.4. Inference-Time Output Refinement

This stage adds inference-time refinements on top of the shared GILT backbone. We use them to improve prediction-time robustness and, for link prediction, inject structural cues that are known to matter. Our main refinement is test-time augmentation (TTA), motivated by the strong empirical benefits of ensemble in tabular foundation models such as TabPFN (Hollmann et al., 2025). Specifically, we apply TTA across node, link, and graph classification by constructing views via random feature rotations and averaging their predictions, which improves robustness to noise. For link prediction, prior work has shown that standard MPNNs are limited by the expressive power of 1-WL and therefore do not adequately capture the pairwise information needed to predict a link(Zhang & Chen, 2018). We therefore introduce an MPLP-inspired (Dong et al., 2024) node labeling estimation strategy to strengthen pairwise encoding for link-level tasks, while leaving the shared backbone unchanged.

3.5. Pre-training

GILT is not trained to solve specific tasks but is taught the general meta-skill of in-context learning with **only one unified model**. The goal of pre-training is to optimize its parameters to become an effective few-shot reasoner. **Compared with many previous GFMs, our pre-training corpus is substantially more diverse** (Liu et al., 2024; Zhao et al., 2025; 2024a; Sun et al., 2025a). Specifically, it consists of 22 datasets spanning domains such as citation, social, and molecular networks, totaling over 450,000 nodes and 4 million edges, with individual graph sizes ranging from tens to over 170,000 nodes. The test datasets are completely disjoint from the pre-training datasets. GILT learns via a multi-task objective covering node, link, and graph classification, on features with dimensions varying from single digits to over 8,000. Further details are in Appendix C.1.

At each training step, a few-shot task consisting of a support set $\mathcal{S}$ and a query set $\mathcal{Q}$ is generated from our diverse training corpus. This task is then formatted and passed through the GILT architecture to produce predictions. A standard cross-entropy loss is computed between these predictions and the ground-truth labels of the query items, y_j:

$$\mathcal{L} = -\frac{1}{|\mathcal{Q}|} \sum_{x_j \in \mathcal{Q}} \log(P(y = y_j | x_j)) \tag{5}$$

This loss is then backpropagated to update all learnable parameters in the model. By optimizing the model over millions of such task instances, the framework is not trained to memorize specific graphs or labels. Instead, it is explicitly trained to learn the meta-skill of **inferring a task's rules from a given support set and applying them to a query set**, thereby acquiring its ability to perform in-context generalization on completely unseen graphs.

4. Experiment

We conduct comprehensive experiments to evaluate GILT across three fundamental graph learning tasks: node classification, link prediction, and graph classification. The core objective is to assess its few-shot performance on unseen graphs against a suite of contemporary GFMs.

4.1. Experiment Setup

Datasets. To ensure a fair comparison, we selected datasets that are canonical for their respective tasks in the literature.

For **node classification**, we use the widely-cited `Cora`, `Citeseer`, `Pubmed` (Yang et al., 2016), and `WikiCS` (Mernyei & Cangea, 2020) benchmarks. For **link prediction**, we again use the Planetoid datasets and add the `ogbl-collab` (Hu et al., 2020a) benchmark for large-scale evaluation. For **graph classification**, we employ standard OGB benchmarks (Hu et al., 2020a), `ogbg-molhiv` and `ogbg-molpcba`, which are popular benchmarks for this task. Crucially, the features in these standard benchmarks are **not natural language**, but high-dimensional numerical features. The statistics are shown in Appendix A.

Baselines We compare GILT against a comprehensive set of baselines representing key paradigms. The competitive landscape differs significantly for each of our three target tasks, so we outline our comparison strategy individually:

For **Node Classification**, prior benchmarks are relatively fragmented and often rely on inconsistent evaluation protocols. To ensure a fair comparison, **we therefore re-evaluate all applicable baselines under a unified few-shot setting**. Our comparisons include standard supervised and self-supervised baselines (MLP, GCN (Kipf & Welling, 2017), GAT (Velickovic et al., 2018), DGI (Velickovic et al., 2019), and GraphCL (You et al., 2020)) as strong general-purpose anchors, few-shot GFM baselines that adapt through either tuning (GCOPE (Zhao et al., 2024a), RiemannGFM (Sun et al., 2025a), and MDGFM (Wang et al., 2025b)) or in-context inference (OFA (Liu et al., 2024) and GraphAny (Zhao et al., 2025)), and LLM-based methods such as LLaga (Chen et al., 2024), ZeroG (Li et al., 2024b), and GOFA (Kong et al., 2025). We treat these LLM-based methods as broader contextual references rather than strictly matched few-shot competitors, since they rely on textual node and class descriptions unavailable in our setting. For **Link Prediction**, few-shot graph foundation model baselines are extremely scarce: UniLP (Dong et al., 2025) is the only link-level GFM baseline we were able to identify for clean reproduction. Standard GNN link predictors under a strict 5-shot protocol collapse to near-random guessing, so they do not provide a meaningful few-shot reference. We therefore compare GILT primarily against fully supervised link prediction methods, including standard GNNs (GCN (Kipf & Welling, 2016), GraphSAGE (Hamilton et al., 2017)) and specialized methods (SEAL (Zhang & Chen, 2018), MaskGAE (Li et al., 2023)) trained with full training supervision. This fully supervised setting is substantially easier than ours, since those models are optimized in-task with many labeled edges. For **Graph Classification**, we compare against available few-shot GFM baselines, including OFA (Liu et al., 2024) and GFT (Wang et al., 2024c), as well as standard supervised GNNs such as GCN (Kipf & Welling, 2017) and GAT (Velickovic et al., 2018). Importantly, **GILT is the only method evaluated in a unified *cross-task* regime (node, link, and graph) with one shared framework, while all baselines are evaluated *in-task* with task-specific setups.** This makes our setting **strictly harder.** Details for baseline evaluation are shown in Appendix C.2.

Evaluation Protocol We adhere to a strict evaluation protocol to prevent data leakage. The support set is sampled from the training split, and the query set for final evaluation is sampled from the test split. We report `Accuracy` for node classification, `Hits@K` for link prediction and `ROC-AUC` for graph classification. On the `Planetoid` datasets, following prior work, we create a random 70%/10%/20% train/valid/test split and report `Hits@100`; the negative pool for Hits@100 is sampled uniformly and is matched in size to the test positive split. All other evaluations use the official public data splits where available.

4.2. Performance and Efficiency Analysis

In this section, we present the main results, focusing on GILT's two main strengths: its state-of-the-art few-shot performance across tasks and its superior inference efficiency.

Node Classification. As shown in Table 2, GILT sets a new state of the art in few-shot node classification, achieving **the highest average accuracy** in both 1-shot (53.15%) and 5-shot (69.51%) settings, with leading results on most dataset-shot combinations. Importantly, this gain is obtained under **a significantly stronger setting: GILT is text-free, requires no test-time tuning, and operates as a unified model across node, link, and graph tasks.** Even against strong task-specific baselines, GILT still delivers superior overall performance, highlighting the effectiveness of its unified in-context learning framework. Another instructive pattern is that **classical supervised and self-supervised GNN baselines outperform several tuning-based GFMs** on many settings under our unified evaluation. We attribute this to the high sensitivity of tuning-based adaptation in the extreme few-shot regime: methods that rely on per-task gradient updates are particularly vulnerable to changes in protocol and target datasets, and therefore do not always transfer cleanly from their original evaluation settings to our unified benchmark. At the same time, the strong performance of these re-evaluated classical baselines suggests that earlier results for standard GNNs were partly constrained by suboptimal hyperparameter choices for few-shot tasks; once a more reasonable combination of hyperparameter is chosen, they become much stronger anchors than is often assumed. In contrast, GILT avoids this fragility altogether, as its in-context adaptation requires no downstream tuning, making it substantially more robust and practical.

Link Prediction. As shown in Table 3a, GILT delivers strong few-shot link prediction performance. Compared with fully supervised baselines, which operate in a substantially easier setting than ours,**GILT still outperforms these baselines on Cora, Citeseer, and ogbl-collab, and**

Table 2. **Few-shot Node Classification Performance.** We report 1-shot and 5-shot accuracy (%) against state-of-the-art few-shot baselines. GILT is evaluated in a unified cross-task setting.

	Cora		Citeseer		Pubmed		WikiCS		Average	
Model	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Supervised Baselines										
MLP	31.14 ± 4.53	42.54 ± 5.89	26.08 ± 5.17	45.30 ± 3.68	48.86 ± 4.74	62.12 ± 1.68	28.30 ± 4.84	55.07 ± 4.45	33.59	51.26
GCN	47.40 ± 13.37	68.90 ± 5.85	29.56 ± 5.24	58.10 ± 8.82	57.02 ± 4.67	69.88 ± 3.13	37.53 ± 5.98	65.85 ± 3.32	43.00	65.56
GAT	55.12 ± 11.68	69.90 ± 3.14	37.76 ± 8.99	59.08 ± 7.55	60.88 ± 3.80	68.50 ± 4.73	36.11 ± 8.42	$\mathbf{67.35 \pm 2.30}$	47.47	66.21
Self-Supervised Pre-training										
DGI	42.70 ± 7.74	61.44 ± 4.34	32.82 ± 7.85	50.78 ± 6.03	58.78 ± 3.89	69.16 ± 6.51	38.94 ± 2.64	63.04 ± 2.78	43.31	61.11
GraphCL	37.18 ± 8.46	63.32 ± 4.13	33.08 ± 6.13	52.24 ± 3.56	57.58 ± 4.18	68.20 ± 4.90	36.38 ± 8.79	63.68 ± 2.37	41.06	61.86
Tuning-Based GFMs										
RiemannGFM	25.08 ± 9.52	46.82 ± 15.73	27.22 ± 9.21	36.52 ± 13.56	44.36 ± 8.48	57.28 ± 6.68	49.85 ± 3.30	53.21 ± 4.20	36.63	48.46
GCOPE	39.05 ± 1.73	67.06 ± 1.41	$\mathbf{55.87 \pm 0.23}$	63.90 ± 1.37	40.89 ± 1.22	64.34 ± 1.94	38.72 ± 0.40	47.73 ± 1.29	43.63	60.76
MDGFM	38.80 ± 8.99	60.94 ± 2.96	30.26 ± 8.65	57.88 ± 9.50	50.64 ± 10.78	65.90 ± 2.87	OOM	OOM	-	-
ICL Models										
OFA	30.52 ± 0.62	41.30 ± 1.89	40.86 ± 0.17	52.01 ± 1.12	31.07 ± 0.85	37.70 ± 0.66	38.50 ± 1.12	49.23 ± 0.50	35.24	45.06
GraphAny	49.30 ± 5.95	72.68 ± 2.47	42.66 ± 8.30	62.08 ± 4.98	56.48 ± 8.98	69.54 ± 2.75	$\mathbf{51.12 \pm 7.79}$	57.86 ± 9.53	49.89	65.54
GILT	$\mathbf{56.36 \pm 7.67}$	$\mathbf{73.22 \pm 3.80}$	48.35 ± 4.18	$\mathbf{66.17 \pm 1.78}$	$\mathbf{61.74 \pm 6.03}$	$\mathbf{71.86 \pm 1.55}$	46.13 ± 1.82	66.80 ± 2.38	**53.15**	**69.51**

Table 3. Few-shot performance on link prediction and graph classification. GILT is evaluated in a unified cross-task setting.

(a) Link prediction performance (Hits@K %). Fully supervised baselines are trained on the train split, while others are evaluated in the 5-shot setting.

Model	**Cora**	**Citeseer**	**Pubmed**	**ogbl-collab**
GCN (Supervised)	66.79 ± 1.65	67.08 ± 2.94	53.02 ± 1.39	44.75 ± 1.07
SAGE (Supervised)	55.02 ± 4.03	57.01 ± 3.74	39.66 ± 0.72	48.10 ± 0.81
SEAL (Supervised)	81.71 ± 1.30	83.89 ± 2.15	75.54 ± 1.32	64.74 ± 0.43
MaskGAE (Supervised)	82.48 ± 0.76	86.11 ± 2.26	$\mathbf{80.38 \pm 0.82}$	65.84 ± 0.47
UniLP (5-shot)	52.91 ± 4.62	50.40 ± 7.49	54.44 ± 1.34	58.00 ± 3.10
GILT (5-shot)	$\mathbf{85.29 \pm 2.46}$	$\mathbf{87.68 \pm 0.91}$	72.30 ± 2.00	$\mathbf{67.83 \pm 0.34}$

(b) Graph classification performance in AUC under a 5-shot setting.

Model	**ogbg-molhiv**	**ogbg-molpcba**
GCN (5-shot)	56.23 ± 6.98	58.43 ± 1.47
GAT (5-shot)	56.36 ± 7.22	51.60 ± 0.86
OFA (5-shot)	49.05 ± 1.37	51.01 ± 0.57
GFT (5-shot)	52.78 ± 7.51	$\mathbf{59.67 \pm 1.41}$
GILT (5-shot)	$\mathbf{58.17 \pm 8.87}$	58.83 ± 0.79

achieves the best results on these three benchmarks. Pubmed remains the main exception, where fully supervised specialist training retains an advantage. These gains indicate that GILT has learned highly generalizable patterns of link formation, which can be rapidly activated by in-context learning from only a few support examples.

Graph Classification. The framework's strong performance extends to the difficult task of few-shot graph classification on challenging molecular benchmarks (Table 3b). Under this 5-shot setting, GILT achieves a ROC-AUC of 58.17% on `ogbg-molhiv` and 58.83% on `ogbg-molpcba`. It is the strongest method on `ogbg-molhiv`, and on `ogbg-molpcba` it remains highly competitive, outperforming GCN, GAT, and OFA while trailing only GFT (Wang et al., 2024c) which needs dataset specific tuning. These results show that GILT transfers effectively to graph-level tasks even on challenging molecular benchmarks. In particular, its strong performance on the especially large and diverse `ogbg-molpcba` dataset is achieved with a single unified, tuning-free, cross-task backbone rather than a graph-classification-specific training pipeline. Overall, these results highlight both the difficulty of graph-level transfer and the versatility of GILT as a multi-task foundational model.

Efficiency Analysis. A critical advantage of GILT's tuning-free paradigm is its efficiency, visualized in Figure 2. For non-tuning-based methods, the reported time is the total inference time on the Cora test split. For tuning-based methods, we include the dataset-specific tuning time on Cora, since it is part of their deployment. All timings are measured on the same machine using a single NVIDIA RTX 4090 GPU. The plot reveals a stark performance gap across different paradigms. Tuning-free models, such as GILT and GraphAny, are clustered within the sub-second region, demonstrating near-instantaneous response. In contrast, traditional supervised methods like GCN and GAT, while established, still require several seconds for inference. Quantitatively, **GILT achieves a speedup of approx-**

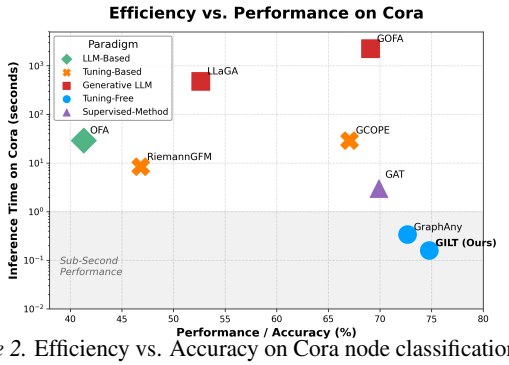

Figure 2. Efficiency vs. Accuracy on Cora node classification. The y-axis is the measured time and the x-axis is accuracy. All models are 5-shot, except for the LLM-based zero-shot baselines.

Table 4. Ablation study results showing 5-shot accuracy (%).

Model Variant	Cora	Citeseer	Pubmed	WikiCS
GILT (Full Model)	**73.22 ± 3.80**	66.17 ± 1.78	**71.86 ± 1.55**	**66.80 ± 2.38**
Ablations on the ICL Transformer				
w/o ICL Transformer	13.00 ± 0.00	7.70 ± 0.00	18.00 ± 0.00	2.51 ± 0.00
w/ Full Token for Prediction	72.97 ± 3.38	66.50 ± 2.02	70.19 ± 2.23	57.19 ± 3.03
Ablations on the Graph Encoder				
w/o Graph Encoder	57.50 ± 7.07	52.31 ± 1.78	60.41 ± 1.10	36.11 ± 1.89
w/ Non-linear GCN	70.76 ± 0.93	61.03 ± 2.50	69.06 ± 3.03	50.98 ± 1.78
w/ 2-layer Encoder	70.52 ± 2.59	**67.40 ± 0.08**	65.96 ± 1.97	55.28 ± 0.93

Table 5. Effect of inference-time refinement.

Node classification				
Variant	**Cora**	**Citeseer**	**Pubmed**	**WikiCS**
base	68.68 ± 4.66	61.90 ± 4.72	71.19 ± 1.73	60.65 ± 0.64
+ TTA	73.22 ± 3.80	66.17 ± 1.78	71.86 ± 1.55	66.80 ± 2.38
Link prediction				
Variant	**Cora**	**Citeseer**	**Pubmed**	**ogbl-collab**
base	85.78 ± 0.82	88.95 ± 0.92	66.52 ± 5.75	56.08 ± 0.65
+ node labeling	85.29 ± 2.46	87.68 ± 0.91	72.30 ± 2.00	67.83 ± 0.34

Table 6. Comparison of GILT's 5-shot performance against LLM-based zero-shot baselines.

Model (Setting)	Cora	Citeseer	Pubmed	WikiCS
LLaGA	52.65 ± 0.57	35.80 ± 0.02	15.84 ± 0.17	52.96 ± 0.30
ZeroG	58.44 ± 5.04	49.32 ± 4.85	68.13 ± 7.86	44.79 ± 10.93
GOFA	69.10 ± 0.08	62.48 ± 0.02	65.91 ± 0.02	66.25 ± 0.41
GILT (5-shot)	**73.22 ± 3.80**	**66.17 ± 1.78**	**71.86 ± 1.55**	**66.80 ± 2.38**

imately 20× over GAT. The efficiency gap widens dramatically when compared to adaptation-heavy tuning-based methods and inference-heavy LLM-based methods. **GILT achieves a speedup of over 180× compared to the tuning-based method GCOPE, and an unprecedented 14000× speedup over the generative LLM approach GOFA.** This underscores the practical scalability of the tuning-free ICL approach and positions GILT as a robust solution for real-world, latency-sensitive applications.

4.3. Architectural Validation and Analysis

To understand the sources of GILT's strong performance, we conduct a series of analyses to validate its key components and deconstruct its in-context learning behavior.

What Architectural Design Makes GILT an Effective In-Context Learner? To identify which components are essential for GILT's in-context learning ability, we perform a series of ablation studies, with results in Table 4.

The ablation results highlight two main findings. First, the **ICL Transformer** is indispensable: removing it entirely (*w/o ICL Transformer*) causes a catastrophic performance collapse across all datasets, while using the full token for prediction leads to a smaller but consistent degradation. Second, the **graph encoder** is also crucial. Removing it causes substantial performance drops, and replacing the deep linear encoder with either a non-linear GCN or a shallow 2-layer encoder generally reduces performance. Overall, these results validate the importance of combining a specialized ICL Transformer with a deep linear graph encoder.

We further isolate the effect of these lightweight inference-time refinements and report them separately in Table 5. Using node classification as a representative example, we find that TTA consistently improves performance, supporting its role as a robust inference-time enhancement in our pipeline. For link prediction, the additional node-labeling component yields large gains on Pubmed and ogbl-collab but small trade-offs on Cora and Citeseer.

How well does GILT learn from in-context examples? To probe the effectiveness of the knowledge GILT acquires from context, we compare it against zero-shot LLMs. These baselines utilize explicit textual descriptions of the classes that often requires laborious pre-processing to obtain, while GILT must infer semantics from the examples alone.

Table 6 show that with an only 5-shot context, GILT's performance clearly surpasses strong LLM baselines with explicit semantic class labels. This highlights a crucial distinction in their underlying mechanisms: while LLMs perform knowledge retrieval by applying vast, pre-existing linguistic information, GILT performs fundamental, graph-native reasoning. It successfully infers a class's functional definition purely from its numerical and structural context.

5. Conclusion

We introduced GILT, a novel Graph Foundational Model designed to be both **LLM-free** and **tuning-free**. Our key innovation is reframing few-shot graph tasks as a token-based reasoning problem, allowing a pre-trained Transformer to learn from examples directly at inference time. Experiments confirm that this in-context learning approach achieves strong few-shot performance, offering a more general and efficient solution than prior methods.

References

Bojchevski, A. and Günnemann, S. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Chamberlain, B. P., Shirobokov, S., Rossi, E., Frasca, F., Markovich, T., Hammerla, N. Y., Bronstein, M. M., and Hansmire, M. Graph neural networks for link prediction with subgraph sketching. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Chen, Q., Wang, L., Zheng, B., and Song, G. Dagprompt: Pushing the limits of graph prompting with a distribution-aware graph prompt tuning approach. In Long, G., Blumestein, M., Chang, Y., Lewin-Eytan, L., Huang, Z. H., and Yom-Tov, E. (eds.), *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pp. 4346–4358. ACM, 2025. doi: 10.1145/3696410.3714917.

Chen, R., Zhao, T., Jaiswal, A. K., Shah, N., and Wang, Z. Llaga: Large language and graph assistant. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

Chen, Z., Mao, H., Li, H., Jin, W., Wen, H., Wei, X., Wang, S., Yin, D., Fan, W., Liu, H., and Tang, J. Exploring the potential of large language models (llms)in learning on graphs. *SIGKDD Explor.*, 25(2):42–61, 2023. doi: 10.1145/3655103.3655110.

Dong, K., Guo, Z., and Chawla, N. V. Pure message passing can estimate common neighbor for link prediction. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Dong, K., Mao, H., Guo, Z., and Chawla, N. V. Universal link predictor by in-context learning on graphs. *Trans. Mach. Learn. Res.*, 2025, 2025.

Fan, W., Wang, S., Huang, J., Chen, Z., Song, Y., Tang, W., Mao, H., Liu, H., Liu, X., Yin, D., and Li, Q. Graph machine learning in the era of large language models (llms). *CoRR*, abs/2404.14928, 2024. doi: 10.48550/ARXIV.2404.14928.

Fang, T., Zhang, Y., Yang, Y., Wang, C., and Chen, L. Universal prompt tuning for graph neural networks. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Fey, M. and Lenssen, J. E. Fast graph representation learning with pytorch geometric. *CoRR*, abs/1903.02428, 2019.

Hamilton, W. L., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1024–1034, 2017.

He, X., Bresson, X., Laurent, T., Perold, A., LeCun, Y., and Hooi, B. Harnessing explanations: LLM-to-LM interpreter for enhanced text-attributed graph representation learning. In *The Twelfth International Conference on Learning Representations*, 2024.

He, Y., Sui, Y., He, X., and Hooi, B. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs. In Sun, Y., Chierichetti, F., Lauw, H. W., Perlich, C., Tok, W. H., and Tomkins, A. (eds.), *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.1, KDD 2025, Toronto, ON, Canada, August 3-7, 2025*, pp. 448–459. ACM, 2025a. doi: 10.1145/3690624.3709277.

He, Y., Sui, Y., He, X., Liu, Y., Sun, Y., and Hooi, B. Unigraph2: Learning a unified embedding space to bind multimodal graphs. In Long, G., Blumestein, M., Chang, Y., Lewin-Eytan, L., Huang, Z. H., and Yom-Tov, E. (eds.), *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pp. 1759–1770. ACM, 2025b. doi: 10.1145/3696410.3714818.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

Hollmann, N., Müller, S., Purucker, L., Krishnakumar, A., Körfer, M., Hoo, S. B., Schirrmeister, R. T., and Hutter, F. Accurate predictions on small data with a tabular foundation model. *Nat.*, 637(8044):319–326, 2025. doi: 10.1038/S41586-024-08328-6.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020a.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V. S., and Leskovec, J. Strategies for pre-training graph neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020b.

Hu, Z., Li, Y., Chen, Z., Wang, J., Liu, H., Lee, K., and Ding, K. Let's ask GNN: empowering large language model for graph in-context learning. In Al-Onaizan, Y., Bansal, M., and Chen, Y. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pp. 1396–1409. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.75.

Huang, Q., Ren, H., Chen, P., Kržmanc, G., Zeng, D., Liang, P., and Leskovec, J. Prodigy: enabling in-context learning over graphs. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *CoRR*, abs/1611.07308, 2016.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Klicpera, J., Bojchevski, A., and Günnemann, S. Predict then propagate: Graph neural networks meet personalized pagerank. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Kong, L., Feng, J., Liu, H., Huang, C., Huang, J., Chen, Y., and Zhang, M. GOFA: A generative one-for-all model for joint graph language modeling. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Li, J., Wu, R., Sun, W., Chen, L., Tian, S., Zhu, L., Meng, C., Zheng, Z., and Wang, W. What's behind the mask: Understanding masked graph modeling for graph autoencoders. In Singh, A. K., Sun, Y., Akoglu, L., Gunopulos, D., Yan, X., Kumar, R., Ozcan, F., and Ye, J. (eds.), *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pp. 1268–1279. ACM, 2023. doi: 10.1145/3580305.3599546.

Li, J., Sun, X., Li, Y., Li, Z., Cheng, H., and Yu, J. X. Graph intelligence with large language models and prompt learning. In Baeza-Yates, R. and Bonchi, F. (eds.), *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 6545–6554. ACM, 2024a. doi: 10.1145/3637528.3671456.

Li, P., Wang, Y., Wang, H., and Leskovec, J. Distance encoding: Design provably more powerful neural networks for graph representation learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Li, Y., Wang, P., Li, Z., Yu, J. X., and Li, J. Zerog: Investigating cross-dataset zero-shot transferability in graphs. In Baeza-Yates, R. and Bonchi, F. (eds.), *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 1725–1735. ACM, 2024b. doi: 10.1145/3637528.3671982.

Lin, M., Hong, X., Li, W., and Lu, S. Unified graph neural networks pre-training for multi-domain graphs. In Walsh, T., Shah, J., and Kolter, Z. (eds.), *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 12165–12173. AAAI Press, 2025. doi: 10.1609/AAAI.V39I11.33325.

Liu, H., Feng, J., Kong, L., Liang, N., Tao, D., Chen, Y., and Zhang, M. One for all: Towards training one graph model for all classification tasks. In *The Twelfth International Conference on Learning Representations*, 2024.

Liu, J., Yang, C., Lu, Z., Chen, J., Li, Y., Zhang, M., Bai, T., Fang, Y., Sun, L., Yu, P. S., and Shi, C. Graph

foundation models: Concepts, opportunities and challenges. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(6): 5023–5044, 2025. doi: 10.1109/TPAMI.2025.3548729.

Liu, Z., Yu, X., Fang, Y., and Zhang, X. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In Ding, Y., Tang, J., Sequeda, J. F., Aroyo, L., Castillo, C., and Houben, G. (eds.), *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pp. 417–428. ACM, 2023. doi: 10.1145/3543507.3583386.

Lu, J., Sun, Y., and Yang, S. In-context time series predictor. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Mao, H., Chen, Z., Tang, W., Zhao, J., Ma, Y., Zhao, T., Shah, N., Galkin, M., and Tang, J. Position: Graph foundation models are already here. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

Mernyei, P. and Cangea, C. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *CoRR*, abs/2007.02901, 2020.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8024–8035, 2019.

Plenz, M. and Frank, A. Graph language models. In Ku, L., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 4477–4494. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.245.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Ren, X., Tang, J., Yin, D., Chawla, N. V., and Huang, C. A survey of large language models for graphs. In Baeza-Yates, R. and Bonchi, F. (eds.), *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 6616–6626. ACM, 2024. doi: 10.1145/3637528.3671460.

Ribeiro, L. F. R., Saverese, P. H. P., and Figueiredo, D. R. *struc2vec*: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pp. 385–394. ACM, 2017. doi: 10.1145/3097983.3098061.

Rozemberczki, B. and Sarkar, R. Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In d'Aquin, M., Dietze, S., Hauff, C., Curry, E., and Cudré-Mauroux, P. (eds.), *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pp. 1325–1334. ACM, 2020. doi: 10.1145/3340531.3411866.

Shchur, O., Mumme, M., Bojchevski, A., and Günnemann, S. Pitfalls of graph neural network evaluation. *CoRR*, abs/1811.05868, 2018.

Sun, L., Huang, Z., Zhou, S., Wan, Q., Peng, H., and Yu, P. S. Riemanngfm: Learning a graph foundation model from riemannian geometry. In Long, G., Blumestein, M., Chang, Y., Lewin-Eytan, L., Huang, Z. H., and Yom-Tov, E. (eds.), *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025-2 May 2025*, pp. 1154–1165. ACM, 2025a. doi: 10.1145/3696410.3714952.

Sun, M., Zhou, K., He, X., Wang, Y., and Wang, X. GPPT: graph pre-training and prompt tuning to generalize graph neural networks. In Zhang, A. and Rangwala, H. (eds.), *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pp. 1717–1727. ACM, 2022. doi: 10.1145/3534678.3539249.

Sun, X., Cheng, H., Li, J., Liu, B., and Guan, J. All in one: Multi-task prompting for graph neural networks. In Singh, A. K., Sun, Y., Akoglu, L., Gunopulos, D., Yan, X., Kumar, R., Ozcan, F., and Ye, J. (eds.), *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pp. 2120–2131. ACM, 2023a. doi: 10.1145/3580305.3599256.

Sun, X., Zhang, J., Wu, X., Cheng, H., Xiong, Y., and Li, J. Graph prompt learning: A comprehensive survey and

beyond. *CoRR*, abs/2311.16534, 2023b. doi: 10.48550/ARXIV.2311.16534.

Sun, Y., Ma, Z., Fang, Y., Ma, J., and Tan, Q. Graphicl: Unlocking graph learning potential in llms through structured prompt design. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pp. 2440–2459. Association for Computational Linguistics, 2025b. doi: 10.18653/V1/2025.FINDINGS-NAACL.131.

Tang, J., Yang, Y., Wei, W., Shi, L., Su, L., Cheng, S., Yin, D., and Huang, C. Graphgpt: Graph instruction tuning for large language models. In Yang, G. H., Wang, H., Han, S., Hauff, C., Zuccon, G., and Zhang, Y. (eds.), *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pp. 491–500. ACM, 2024. doi: 10.1145/3626772.3657775.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Velickovic, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., and Hjelm, R. D. Deep graph infomax. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Wang, D., Zuo, Y., Li, F., and Wu, J. Llms as zero-shot graph learners: Alignment of GNN representations with LLM token embeddings. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a.

Wang, H. P., Liu, S., Wei, R., and Li, P. Generalization principles for inference over text-attributed graphs with large language models. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J. (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025a.

Wang, J., Wu, J., Hou, Y., Liu, Y., Gao, M., and McAuley, J. J. Instructgraph: Boosting large language models via graph-centric instruction tuning and preference alignment. In Ku, L., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pp. 13492–13510. Association for Computational Linguistics, 2024b. doi: 10.18653/V1/2024.FINDINGS-ACL.801.

Wang, S., Wang, B., Shen, Z., Deng, B., and Kang, Z. Multi-domain graph foundation models: Robust knowledge transfer via topology alignment. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J. (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025b.

Wang, X., Wang, W., Cao, Y., Shen, C., and Huang, T. Images speak in images: A generalist painter for in-context visual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 6830–6839. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00660.

Wang, Z., Zhang, Z., Chawla, N. V., Zhang, C., and Ye, Y. Gft: Graph foundation model with transferable tree vocabulary. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 107403–107443. Curran Associates, Inc., 2024c.

Wu, F., Jr., A. H. S., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. Q. Simplifying graph convolutional networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6861–6871. PMLR, 2019.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. S. Moleculenet: A benchmark for molecular machine learning. *CoRR*, abs/1703.00564, 2017.

Xia, L., Kao, B., and Huang, C. Opengraph: Towards open graph foundation models. In Al-Onaizan, Y., Bansal, M., and Chen, Y. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pp. 2365–2379. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.132.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Yang, R., Shi, J., Xiao, X., Yang, Y., Bhowmick, S. S., and Liu, J. PANE: scalable and effective attributed network embedding. *VLDB J.*, 32(6):1237–1262, 2023. doi: 10.1007/S00778-023-00790-4.

Yang, Z., Cohen, W. W., and Salakhutdinov, R. Revisiting semi-supervised learning with graph embeddings. In Balcan, M. and Weinberger, K. Q. (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 40–48. JMLR.org, 2016.

Yang, Z., Han, J., Wang, C., and Liu, H. Graphlora: Structure-aware contrastive low-rank adaptation for cross-graph transfer learning. In Sun, Y., Chierichetti, F., Lauw, H. W., Perlich, C., Tok, W. H., and Tomkins, A. (eds.), *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.1, KDD 2025, Toronto, ON, Canada, August 3-7, 2025*, pp. 1785–1796. ACM, 2025. doi: 10.1145/3690624.3709186.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Yu, X., Gong, Z., Zhou, C., Fang, Y., and Zhang, H. SAMGPT: text-free graph foundation model for multi-domain pre-training and cross-domain adaptation. In Long, G., Blumestein, M., Chang, Y., Lewin-Eytan, L., Huang, Z. H., and Yom-Tov, E. (eds.), *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pp. 1142–1153. ACM, 2025a. doi: 10.1145/3696410.3714828.

Yu, Z., Li, W., Yin, Z., Hong, X., Xiao, S., and Lu, S. Contextual structure knowledge transfer for graph neural networks. In Walsh, T., Shah, J., and Kolter, Z. (eds.), *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 22263–22271. AAAI Press, 2025b. doi: 10.1609/AAAI.V39I21.34381.

Yuan, H., Sun, Q., Shi, J., Fu, X., Hooi, B., Li, J., and Yu, P. S. How much can transfer? BRIDGE: bounded multi-domain graph foundation model with generalization guarantees. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J. (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

Zeng, H., Zhou, H., Srivastava, A., Kannan, R., and Prasanna, V. K. Graphsaint: Graph sampling based inductive learning method. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Zhang, M. and Chen, Y. Link prediction based on graph neural networks. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 5171–5181, 2018.

Zhang, M., Sun, M., Wang, P., Fan, S., Mo, Y., Xu, X., Liu, H., Yang, C., and Shi, C. Graphtranslator: Aligning graph model to large language model for open-ended tasks. In Chua, T., Ngo, C., Kumar, R., Lauw, H. W., and Lee, R. K. (eds.), *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pp. 1003–1014. ACM, 2024a. doi: 10.1145/3589334.3645682.

Zhang, M., Sun, M., Wang, P., Fan, S., Mo, Y., Xu, X., Liu, H., Yang, C., and Shi, C. Graphtranslator: Aligning graph model to large language model for open-ended tasks. In Chua, T., Ngo, C., Kumar, R., Lauw, H. W., and Lee, R. K. (eds.), *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pp. 1003–1014. ACM, 2024b. doi: 10.1145/3589334.3645682.

Zhao, H., Chen, A., Sun, X., Cheng, H., and Li, J. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In Baeza-Yates, R. and Bonchi, F. (eds.), *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 4443–4454. ACM, 2024a. doi: 10.1145/3637528.3671913.

Zhao, J., Jin, D., Ge, M., Shan, L., Wang, X., He, D., and Feng, Z. Fug: Feature-universal graph contrastive pre-training for graphs with diverse node features. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 4003–4034. Curran Associates, Inc., 2024b.

Zhao, J., Zhu, Z., Galkin, M., Mostafa, H., Bronstein, M. M., and Tang, J. Fully-inductive node classification on arbitrary graphs. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Zhu, J., Ding, Z., Yu, J., Tan, J., Li, X., and Qian, W. RELIEF: reinforcement learning empowered graph feature prompt tuning. In Sun, Y., Chierichetti, F., Lauw, H. W.,

Perlich, C., Tok, W. H., and Tomkins, A. (eds.), *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.1, KDD 2025, Toronto, ON, Canada, August 3-7, 2025*, pp. 2159–2170. ACM, 2025a. doi: 10.1145/3690624.3709252.

Zhu, Y., Shi, H., Wang, X., Liu, Y., Wang, Y., Peng, B., Hong, C., and Tang, S. Graphclip: Enhancing transferability in graph foundation models for text-attributed graphs. In Long, G., Blumestein, M., Chang, Y., Lewin-Eytan, L., Huang, Z. H., and Yom-Tov, E. (eds.), *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pp. 2183–2197. ACM, 2025b. doi: 10.1145/3696410.3714801.

Zi, C., Zhao, H., Sun, X., Lin, Y., Cheng, H., and Li, J. Prog: A graph prompt learning benchmark. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

A. Dataset Details

This section provides detailed statistics for the datasets used in our pre-training corpus and for our downstream evaluations. All evaluation datasets were strictly held out and unseen during the pre-training phase.

A.1. Pre-training Datasets

To learn a general-purpose in-context reasoner, GILT was pre-trained on a large and diverse corpus of 22 publicly available graph datasets. This corpus was curated to span multiple domains (social, citation, transportation, web, and molecular) and task levels (node, link, and graph) to ensure the model learns robust and generalizable patterns. A summary of the key datasets included in the pre-training corpus is provided in Table 7.

A.2. Evaluation datasets

We evaluated GILT's few-shot performance on a suite of 8 unseen benchmark datasets. These datasets were chosen as they are standard in the literature and cover all three primary graph learning tasks. Detailed statistics for each evaluation dataset are summarized in Table 8.

A.3. Introductions to the Datasets

To make the benchmark coverage explicit, we provide a brief description of each dataset used in this paper.

Pre-training datasets.

- **ogbn-arxiv** (Hu et al., 2020a): a citation network of arXiv papers, where each node is a paper and labels correspond to subject areas.
- **CS** (Shchur et al., 2018): a co-authorship graph in computer science from the Amazon/Coauthor benchmark collection.
- **Physics** (Shchur et al., 2018): a co-authorship graph in physics from the same benchmark suite.
- **Computers** (Shchur et al., 2018): an Amazon co-purchase graph from the computer products category.
- **Photo** (Shchur et al., 2018): an Amazon co-purchase graph from the photo products category.
- **Flickr** (Zeng et al., 2020): a social graph with user connectivity and profile-level node features.
- **USA** (Ribeiro et al., 2017): a transportation network (airport-style benchmark) where connectivity reflects routes.
- **Brazil** (Ribeiro et al., 2017): a transportation network from Brazil in the same benchmark family.
- **Europe** (Ribeiro et al., 2017): a transportation network from Europe in the same benchmark family.
- **Wiki** (Yang et al., 2023): a web-page graph benchmark with attributed nodes.
- **BlogCatalog** (Yang et al., 2023): a social network benchmark of bloggers with multi-class labels.
- **DBLP** (Bojchevski & Günnemann, 2018): an academic network benchmark from the citation/co-authorship domain.
- **FacebookPagePage** (Rozemberczki & Sarkar, 2020): a social graph where nodes are Facebook pages and edges capture inter-page links.
- **DeezerEurope** (Rozemberczki & Sarkar, 2020): a user social network from the Deezer music platform.
- **LastFMAsia** (Rozemberczki & Sarkar, 2020): a social graph of LastFM users in Asia with country-level labels.
- **bace** (Wu et al., 2017): a molecular property prediction dataset focused on BACE-1 inhibitor activity.
- **bbbp** (Wu et al., 2017): a molecular dataset for blood-brain barrier penetration prediction.
- **tox21** (Wu et al., 2017): a multi-task molecular toxicity benchmark.

Table 7. Statistics of datasets used in the GILT pre-training.

Dataset	Domain	# Graphs	# Nodes	# Edges	# Features	# Classes/Tasks
ogbn-arxiv (Hu et al., 2020a)	Citation	1	169,343	1,166,243	128	40
CS (Shchur et al., 2018)	Co-authorship	1	18,333	163,788	6,805	15
Physics (Shchur et al., 2018)	Co-authorship	1	34,493	495,924	8,415	5
Computers (Shchur et al., 2018)	Co-purchase	1	13,752	491,722	767	10
Photo (Shchur et al., 2018)	Co-purchase	1	7,650	238,162	745	8
Flickr (Zeng et al., 2020)	Social	1	89,250	899,756	500	7
USA (Ribeiro et al., 2017)	Transportation	1	1,190	13,599	1,190	4
Brazil (Ribeiro et al., 2017)	Transportation	1	131	1,038	131	4
Europe (Ribeiro et al., 2017)	Transportation	1	399	5,995	399	4
Wiki (Yang et al., 2023)	Web pages	1	2,405	17,981	4,973	17
BlogCatalog (Yang et al., 2023)	Social	1	5,196	343,486	8,189	6
DBLP (Bojchevski & Günnemann, 2018)	Citation	1	17,716	105,734	1,639	4
FacebookPagePage (Rozemberczki & Sarkar, 2020)	Social	1	22,470	342,004	128	4
DeezerEurope (Rozemberczki & Sarkar, 2020)	Social	1	28,281	185,504	128	2
LastFMAsia (Rozemberczki & Sarkar, 2020)	Social	1	7,624	55,612	128	18
bace (Wu et al., 2017)	Molecular	1,513	34.1 (avg)	73.7 (avg)	9	2
bbbp (Wu et al., 2017)	Molecular	2,039	23.9 (avg)	51.6 (avg)	9	2
tox21 (Wu et al., 2017)	Molecular	7,831	18.6 (avg)	38.6 (avg)	9	12
toxcast (Wu et al., 2017)	Molecular	8,575	18.8 (avg)	38.5 (avg)	9	617
clintox (Wu et al., 2017)	Molecular	1,478	26.1 (avg)	55.8 (avg)	9	2
muv (Wu et al., 2017)	Molecular	93,087	24.2 (avg)	51.2 (avg)	9	17
sider (Wu et al., 2017)	Molecular	1,427	33.6 (avg)	70.7 (avg)	9	27

Table 8. Statistics of datasets used in the GILT evaluation.

Dataset	Domain	# Graphs	# Nodes	# Edges	# Features	# Classes/Tasks
Cora (Yang et al., 2016)	Citation	1	2,708	10,556	1,433	7
Citeseer (Yang et al., 2016)	Citation	1	3,327	9,104	3,703	6
Pubmed (Yang et al., 2016)	Citation	1	19,717	88,648	500	3
WikiCS (Yang et al., 2016)	Web pages	1	11,701	216,123	300	10
ogbl-collab (Hu et al., 2020a)	Co-authorship	1	235,868	1,285,465	128	Link Prediction
ogbg-molhiv (Hu et al., 2020a)	Molecular	41,127	25.5 (avg)	27.5 (avg)	9	1
ogbg-molpcba (Hu et al., 2020a)	Molecular	437,929	26.0 (avg)	28.1 (avg)	9	128

- **toxcast** (Wu et al., 2017): a large-scale multi-task molecular toxicity benchmark with many assay endpoints.
- **clintox** (Wu et al., 2017): a molecular benchmark for clinical trial toxicity and FDA approval-related endpoints.
- **muv** (Wu et al., 2017): a sparse virtual screening benchmark with challenging molecular activity tasks.
- **sider** (Wu et al., 2017): a molecular benchmark for side-effect related prediction tasks.

Evaluation datasets.

- **Cora** (Yang et al., 2016): a classic citation network benchmark for node classification.
- **Citeseer** (Yang et al., 2016): a citation graph benchmark with sparse bag-of-words node features.
- **Pubmed** (Yang et al., 2016): a large biomedical citation network benchmark.
- **WikiCS** (Mernyei & Cangea, 2020): a Wikipedia-based benchmark for node classification on web pages.
- **ogbl-collab** (Hu et al., 2020a): a collaboration graph benchmark for link prediction.
- **ogbg-molhiv** (Hu et al., 2020a): a molecular graph benchmark for HIV activity prediction.
- **ogbg-molpcba** (Hu et al., 2020a): a large molecular graph benchmark with many bioassay tasks.

B. Baseline Introductions

In this section, we briefly introduce all baseline methods used in our experiments. These baselines can be classified into four families: tuning-based methods, in-context learning methods, LLM-based methods, and supervised baselines.

B.1. Tuning-Based Methods

GCOPE (Zhao et al., 2024a). GCOPE operates by projecting disparate graph features into a shared dimension, typically through SVD (singular value decomposition). The mechanism introduces virtual coordinator nodes that form fully connected subnetworks with internal nodes and establish inter-coordinator edges to bridge isolated domains. This architecture creates a joint adjacency matrix intended to unify independent graph distributions into a cohesive system. Training employs a composite objective that combines graph contrastive learning with an auxiliary feature reconstruction loss, aiming to preserve salient data characteristics while distilling transferable representations. Downstream adaptation is achieved by converting target tasks into graph-level problems via induced subgraphs, followed by standard finetuning or the application of graph prompts.

RiemannGFM (Sun et al., 2025a). RiemannGFM is a graph foundation model that utilizes a structural vocabulary of trees and cycles to learn universal graph representations. The mechanism is built on a product bundle that integrates Riemannian manifolds for local geometry and tangent spaces for global structural encodings. Its layers comprise a vocabulary learning module that employs cross-geometry attention to update node coordinates in hyperbolic and hyperspherical manifolds. Concurrently, a global learning module utilizes bundle convolution and parallel transport to aggregate node encodings while addressing tangent space incompatibility. Pretraining is conducted via geometric contrastive learning between the hyperbolic and hyperspherical views within a shared tangent space. Downstream tasks are performed by training a classification head on the generated node encodings.

MDGFM (Wang et al., 2025b). MDGFM addresses cross-domain structural discrepancies by implementing a unified framework centered on topology-aware alignment. The architecture utilizes a decoupled embedding mechanism where an adaptive balance token dynamically weighs node features against aggregated topological information. To mitigate inherent noise and align disparate structures, the model integrates a graph structure learning module that refines source graphs and extracts domain-invariant knowledge. Pre-training is driven by a contrastive learning objective that maximizes mutual information between original and refined graph representations to ensure the preservation of global structural properties. For downstream adaptation, MDGFM employs a dual-stream prompt strategy—combining a meta prompt to model inter-domain relationships with a task-specific prompt for precise target alignment—facilitating robust knowledge transfer even in few-shot scenarios.

GFT (Wang et al., 2024c). GFT operates by identifying transferable structural patterns as computation trees derived from the graph message-passing process. These computation trees are utilized as tokens within a shared structural vocabulary, allowing the model to encode generalizable patterns across disparate domains and tasks. The mechanism unifies node-, edge-, and graph-level classification by extracting task-relevant subtrees as basic learning instances for pretraining. For downstream tasks, the model undergoes supervised fine-tuning where the entire parameter set is updated using target-specific subtrees to align the learned vocabulary with downstream labels.

B.2. In-Context Learning Methods

OFA (Liu et al., 2024). OFA is a graph foundation model that unifies diverse graph data by representing nodes, edges, and tasks using a standardized natural language description framework. The mechanism utilizes a Graph-to-Text approach where graph structural information and metadata are converted into textual prompts, which are then processed by an LLM backbone. By aligning disparate feature spaces into a common linguistic embedding space, OFA enables a single model to handle multiple graph domains and tasks simultaneously. For downstream tasks, the model employs a prompt-based strategy where task-specific natural language instructions are appended to the input, allowing the model to adapt to new domains through in-context learning.

GraphAny (Zhao et al., 2025). GraphAny is a graph foundation model designed for node classification that applies an analytical solution to generalize to any target graph without requiring training on that specific dataset. Its mechanism utilizes a set of graph filters to produce multiple candidate predictions, which are then adaptively combined using weights produced by a lightweight meta-network based on the graph's structural and feature statistics. For downstream tasks, the model identifies the optimal combination of its internal filters and solve an analytical optimization problem using the provided few-shot labels.

UniLP (Dong et al., 2025). UniLP is a universal link prediction foundation model that leverages in-context learning to generalize across diverse graph datasets without task-specific fine-tuning. Its mechanism addresses the challenge of conflicting connectivity patterns by utilizing a set of in-context links sampled from the target graph as reference. These examples, along with the query link, are processed through a shared subgraph GNN encoder and an attention-based fusion module, allowing the model to adaptively recognize the specific structural logic of the current graph. For downstream tasks, UniLP enables zero-shot transfer by simply providing a few in-context examples during inference, achieving performance competitive with or superior to fully

supervised, domain-specific models.

B.3. LLM-Based Methods

LLaGA (Chen et al., 2024). LLaGA reformulates graph learning by mapping structural information into the input space of LLMs. The framework reorganizes graph nodes and their local neighborhoods into structure-aware sequences, which are then projected into the LLM's token embedding space via a versatile projector. This approach allows the model to leverage the pre-trained reasoning capabilities of LLMs to interpret complex graph topologies and node attributes simultaneously. LLaGA demonstrates strong generalizability by performing various graph tasks, such as node classification and link prediction, in both supervised and zero-shot settings across diverse domains without requiring task-specific structural modifications.

ZeroG (Li et al., 2024b). ZeroG addresses cross-dataset zero-shot transferability by utilizing LLMs to unify heterogeneous feature spaces and label sets. The framework employs an LLM-based encoder to map diverse node attributes and class semantics into a consistent embedding space, mitigating the challenge of feature misalignment across domains. To capture structural dependencies without requiring task-specific finetuning, ZeroG introduces a prompt-based subgraph sampling module and a relation-aware GNN that generalizes across varied graph topologies. During inference, it performs zero-shot classification by computing the similarity between node-centric subgraph representations and potential class semantic embeddings, enabling effective generalization to unseen datasets and label spaces.

GOFA (Kong et al., 2025). GOFA implements a generative one-for-all framework by interleaving graph-aware layers with a frozen LLM to achieve joint graph-language modeling. The architecture incorporates a symmetry-preserving GNN that extracts structural representations, which are subsequently integrated into the LLM's transformer blocks through a cross-attention mechanism. This design enables the model to process arbitrary graph topologies and textual attributes within a unified generative decoder, supporting a wide range of tasks through natural language prompting. Pre-trained on diverse graph datasets via a next-token prediction objective, GOFA exhibits strong in-context learning and zero-shot generalization capabilities across various graph-structured domains.

B.4. Supervised Baselines

GCN (Kipf & Welling, 2017). GCN is a standard message-passing baseline which updates node representations via the propagation rule $H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$, utilizing a normalized adjacency matrix $\tilde{A}$ with added self-loops and its degree matrix $\tilde{D}$. For downstream tasks, the model is paired with a task-specific head and undergoes end-to-end supervised training where all parameters are updated on target data.

GraphSAGE (Hamilton et al., 2017). GraphSAGE is a graph neural network that utilizes layer-wise message passing and neighborhood aggregation to generate node embeddings. For downstream tasks, the model typically undergoes supervised end-to-end training where its parameters are iteratively updated to minimize a specific objective function.

GAT (Velickovic et al., 2018). GAT introduces an attention-based neighborhood aggregation mechanism that assigns non-uniform weights to neighbors using coefficients $a_{ij} = \text{softmax}_j(e_{ij})$, where e_{ij} represents the importance of node j to node i. This enables the model to focus on the most relevant parts of the local structure without requiring pre-normalized graph statistics. For downstream adaptation, the attention layers and task-specific heads are typically optimized through supervised finetuning on the target domain.

SEAL (Zhang & Chen, 2018) SEAL is a framework for link prediction that formulates the task as a subgraph classification problem. Its mechanism extracts h-hop enclosing subgraphs for each target link and applies a Double Radius Node Labeling (DRNL) scheme to encode the structural roles of nodes relative to the link's endpoints. These labeled subgraphs are then processed by a graph neural network to learn structural signatures that signify link existence. For downstream tasks, the model undergoes supervised training on these extracted subgraphs to map local structural patterns to link presence labels.

MaskGAE (Li et al., 2023) MaskGAE employs an asymmetric encoder-decoder architecture to perform masked graph modeling. The mechanism involves masking a portion of the input graph edges to reduce redundancy between contrastive subgraph views during training. While a GNN encoder processes only the visible, unmasked subgraph, the decoder utilizes a structure module for edge reconstruction and a degree module for auxiliary node-level degree regression. The framework optimizes a joint objective that combines binary cross-entropy for structural reconstruction with mean squared error for degree prediction.

B.5. Self-Supervised Pretraining Baselines

DGI (Velickovic et al., 2019). DGI is a self-supervised approach that learns node representations by maximizing the mutual information between local patch representations and a global graph summary. It employs a contrastive objective where the model learns to distinguish between real node-

graph pairings and corrupted alternatives generated through a stochastic shuffling mechanism. By forcing the encoder to capture high-level structural properties that are consistent across the entire graph, DGI produces versatile embeddings that can be utilized for various downstream tasks without the need for manual labels.

GraphCL (You et al., 2020) GraphCL is a general contrastive learning framework for GNNs that learns unsupervised node or graph representations by maximizing the agreement between different augmented views of the same graph. The framework utilizes four types of graph augmentations—node dropping, edge perturbation, attribute masking, and subgraph extraction—to create diverse correlated views that force the model to learn invariant structural patterns. By employing a contrastive loss in the latent space, GraphCL enables the extraction of robust features that are transferable to various downstream tasks, such as graph classification, without requiring any manual annotations during the pre-training phase.

C. Implementation Details

This section provides the specific implementation details for our model, GILT, and all baselines used in the experiments.

C.1. GILT Model and Pre-training

Architecture Details. The GILT model evaluated in our experiments was configured with the following architecture. The non-parametric encoder projects all node features into a unified dimension of $d = 128$. The structural encoder is a 6-layer linear GCN, with each aggregation step followed by a LayerNorm with learnable affine parameters. The ICL Transformer consists of 2 layers, with 4 attention heads per layer and a hidden dimension of $d = 1024$ in the feed-forward networks.

More Design Choices. Below we summarize several implementation details and fixed design choices used in the final system.

- Feature Pre-processing: Our feature alignment pipeline is a multi-step process designed to robustly handle graphs with diverse feature dimensions. For graphs with high-dimensional features, we use SVD for dimensionality reduction. For graphs with low-dimensional features, we apply SVD while keeping the same dimensionality and then perform zero padding to match the final model dimension. Finally, the entire processed feature matrix undergoes a crucial column-wise standard scaling, which normalizes each feature dimension independently and prevents padded zeros from distorting the feature statistics.

- Graph Encoder: Within the linear GCN encoder, we found two important, and one counter-intuitive, results. First, including the learnable affine parameters in each LayerNorm step was essential for performance; removing them caused a significant loss. Second, we found that adding residual connections between the linear GCN layers, a standard practice in deep networks, was not effective and did not improve results.

- Prototype Formulation: For generating the class prototypes, we confirmed that L2-normalizing the vectors after mean pooling is highly beneficial for stabilizing the model. We also experimented with enforcing an additional orthogonality constraint on the prototypes but found it to be less effective than simple normalization.

- Test-time augmentation: As described in the main text, we apply test-time augmentation across node, link, and graph classification by constructing five transformed views and averaging the resulting predictions. We use this as a general inference-time robustness mechanism rather than a task-specific module.

- Link prediction structural encoding: As described in the main text, we additionally use an MPLP-inspired (Dong et al., 2024) node labeling estimation strategy for link prediction. Conceptually, this can be viewed as a count-based distance encoding of the local common neighborhood around each candidate edge, in the spirit of prior distance-based node labeling schemes such as DRNL and the broader distance-encoding view of structural representation learning (Zhang & Chen, 2018; Li et al., 2020). Concretely, it counts how many shared neighbors fall into a small set of canonical relative-distance encodings with respect to the two endpoints, such as $(1, 1)$, $(2, 1)$, $(1, 2)$, $(1, \infty)$, $(\infty, 1)$, and $(2, 2)$. We estimate these counts using the MPLP estimator, yielding a compact structural summary without explicit enclosing-subgraph extraction.

Pre-training. The pre-training process was conducted for a total of 50 epochs, with each epoch iterating through all tasks sampled from our pre-training corpus. To improve the model's ability to learn from varying amounts of context, we employed a shot decay schedule, where the number of shots, was gradually decayed from an initial 20 down to 5 over the course of training. To enhance model robustness, we applied two forms of data augmentation: feature dropout and edge dropout. The final multi-task loss is a weighted sum of the individual task losses, with a hyperparameter controlling their relative importance. For link prediction tasks specifically, the support set is constructed with a balanced 1:1 ratio between positive and negative edges, while the overall negative sampling ratio used for pre-training remains 3:1.

C.2. Baseline Setup

For our experiments, we made a distinction between re-evaluating baselines and citing established results. Specifically, for the node classification task, we conducted a fresh evaluation of all Tuning-Based and ICL baselines using their official public codebases to ensure a direct and fair comparison under our strict few-shot protocol. For all other results, including baselines on other tasks and the supervised models, performance is reported from their original publications or other well-established literature to ensure consistency with community standards.

Supervised Models. For MLP, GCN, and GAT, we implemented lightweight supervised baselines under strict few-shot evaluation protocol. All models were trained from scratch on the few-shot samples only, with early stopping based on validation performance, using dropout of 0.5, learning rate of 0.01, weight decay of 5×10^{-4} and hidden dimension of 128 for MLP and GCN. We report the mean and standard deviation over five random seeds.

Self-Supervised Pretraining Models. For DGI and GraphCL, we used a self-supervised pretraining followed by supervised few-shot adaptation protocol. Both methods first pretrained a GCN backbone on ogbn-arxiv, and then transferred the pretrained backbone to each downstream dataset. During adaptation, the model was fine-tuned on the few-shot samples only, with early stopping based on validation performance. For DGI, we used a standard corruption-based graph infomax objective; for GraphCL, we used feature masking and edge dropout to construct two graph views and optimized an InfoNCE contrastive loss. We used hidden dimension 128, dropout of 0.5, pretraining learning rate of 0.001, fine-tuning learning rate of 0.01, and fine-tuning weight decay of 5×10^{-4}. We report the mean and standard deviation over five random seeds.

Tuning-Based Models. For GCOPE (Zhao et al., 2024a), we used its official implementation and default parameters. To ensure a strict separation between pre-training and evaluation data, we modified its pre-training corpus to exclude the Planetoid datasets. During few-shot adaptation, we followed standard procedure by freezing the GNN backbone and only tuning the prompt module. For RiemannGFM (Sun et al., 2025a), while also using its default parameters, we observed that its original prediction mechanism utilizes external class information. To ensure a fair comparison focused solely on the ability to learn from the provided examples, we replaced its final head with a simple linear classifier that was then tuned on the few-shot support set.

ICL Models. For OFA (Liu et al., 2024) and GraphAny (Zhao et al., 2025), we utilized their official public codebases to evaluate them in our few-shot setting. For OFA, we used the default model parameters and its standard Sentence Transformer for generating text embeddings; the checkpoint was pre-trained on a corpus including ogbn-arxiv. For GraphAny, we also used its default parameters and employed the official model checkpoint pre-trained on the ogbn-arxiv dataset for all node classification evaluations.

LLM-Based Models. For LLaGA (Chen et al., 2024), we used the official implementation and followed its hop-token node classification setting. Since the released checkpointsare not trained under the exact zero-shot transfer setting required by our evaluation, we reproduced the projector training ourselves. To avoid target-dataset leakage, we trained the graph-language projector only on source citation graphs and excluded the evaluation dataset from the training corpus. For the graph features, we used SBERT embeddings and generated the required multi-hop propagated embeddings locally, rather than using the released SimTeG feature package, which depends on additional external encoders. For ZeroG (Li et al., 2024b), we used the official codebase and reproduced the citation-domain zero-shot setting ourselves, since the released repository does not directly provide the exact training protocol used in our evaluation. We trained on source citation graphs only, excluding the target dataset in each run. For GOFA (Kong et al., 2025), we used the official implementation and the released instruction-tuned checkpoint. For datasets not directly exposed by the GOFA evaluation scripts, such as CiteSeer, we added TAGLAS-compatible dataset wrappers using the data from (Wang et al., 2025a) and kept the original public test split for zero-shot evaluation. For PubMed, full-test evaluation with GOFA required more than 24 hours, so we reported results on a uniformly sampled subset of 10,000 test nodes under the same inference configuration.

Link Prediction Baselines. Few-shot graph foundation model baselines for link prediction are extremely limited. In practice, UniLP (Dong et al., 2025) is the only dedicated link-level few-shot GFM baseline we were able to identify and evaluate. We do not report 5-shot results for standard supervised GNN link predictors, because under our protocol they degenerate to near-random guessing and therefore fail to provide a meaningful few-shot reference. We therefore compare mainly against fully supervised link prediction baselines trained with the full training split, which is a substantially easier setting than ours because those methods are optimized directly on the target graph with much richer supervision. For these fully supervised baselines, the performance of GCN (Kipf & Welling, 2017), GraphSAGE (Hamilton et al., 2017) and SEAL (Zhang & Chen, 2018) is reported from (Chamberlain et al., 2023), since it is a widely recognized and authoritative benchmark for link prediction. We therefore use its published results directly rather than in-

Table 9. Comparison of the reported pre-training corpora of GILT and prior GFMs based on information disclosed in the original papers.

Model	Reported Corpus	Task Coverage	Domain Coverage
GILT	22 datasets	Node, link, graph	Citation, social, transportation, web, molecular
OFA (Liu et al., 2024)	3 datasets	Node, link, graph	Citation, knowledge graphs, molecular
GraphAny (Zhao et al., 2025)	1 dataset	Node	Citation
GCOPE (Zhao et al., 2024a)	9 datasets	Node	Citation, e-commerce, web
RiemannGFM (Sun et al., 2025a)	3 datasets	Node, link	Citation, e-commerce, co-authorship

troducing additional implementation and tuning differences on our side. We evaluate MaskGAE (Li et al., 2023) under its original setting, since its paper reports AUC on the Planetoid datasets. For UniLP, we keep its setup as faithful as possible to the original method, changing only what is necessary to adapt evaluation to our 5-shot protocol and to report Hits@100 on the Planetoid datasets.

Graph Classification Baselines. For the graph classification baselines, we evaluate GCN (Kipf & Welling, 2017) and GAT (Velickovic et al., 2018) using the same lightweight supervised 5-shot protocol as in node classification: both models are trained from scratch on the provided few-shot graphs only, with the same optimization setup and early stopping strategy. We evaluate OFA (Liu et al., 2024) in the same 5-shot setting while keeping all other configurations unchanged. More broadly, the relatively weak performance of OFA and GFT under our protocol is consistent with the same mismatch discussed for node classification: these methods rely on support-induced prompt or structural templates whose inferred class semantics become unstable under strict few-shot transfer. For GFT (Wang et al., 2024c), its original implementation pre-trains on datasets that include downstream datasets and selects fine-tuning samples separately from the few-shot setting. To ensure a fair comparison, we exclude the target test dataset from its pre-training corpus and restrict fine-tuning to the provided few-shot examples. Both methods are trained only for graph classification.

C.3. Computational Resources

All experiments were conducted on a Linux server equipped with 8 NVIDIA RTX 4090 GPUs and dual Intel(R) Xeon(R) Platinum 8370C CPUs @ 2.80GHz (128 logical CPUs in total). Unless otherwise specified, the efficiency measurements in Figure 2 were obtained on the same machine using a single NVIDIA RTX 4090 GPU. Our implementation is built using PyTorch (Paszke et al., 2019) and PyTorch Geometric (PyG) (Fey & Lenssen, 2019).

Efficiency Measurement Protocol. To make the efficiency comparison in Figure 2 as transparent as possible, we measured all methods on the Cora node classification benchmark under the same hardware setting using a single RTX 4090 GPU. For non-tuning-based methods, the reported time is the end-to-end inference time on the Cora test split. For tuning-based methods, we additionally include the required dataset-specific adaptation time on Cora before test inference, since this cost is part of their practical deployment and is typically more substantial than the lightweight training used by simple supervised baselines such as GCN. We do not include dataset I/O time in the measurement, since it depends on the storage environment rather than the method itself. In contrast, dataset preprocessing time is included whenever it is part of the method's standard pipeline. No multi-GPU parallelism was used for this comparison. We therefore interpret the reported time as the total wall-clock cost needed to obtain predictions on the target benchmark under each method's standard usage mode.

D. Comparison of Pre-training Corpora

To contextualize the stronger transfer ability of GILT within the graph domain, it is useful to compare its pre-training corpus against those reported for prior GFMs not only in raw scale, but also in task coverage, domain diversity and modality assumptions. Table 9 summarizes these differences in a compact and citation-friendly format, with the comparison restricted to directly relevant graph-domain methods.

Table 9 highlights three patterns that help explain GILT's transfer performance within the graph-domain setting. First, GILT is pre-trained on a substantially larger corpus than the compared prior GFMs, covering 22 datasets, whereas the reported corpora of OFA, GCOPE, and RiemannGFM range from 3 to 9 datasets and GraphAny reports pre-training on a single dataset. Within the current graph-domain literature, this broader scale exposes GILT to a wider variety of feature spaces, graph structures, and task formulations during pre-training.

Second, GILT combines corpus scale with broader task coverage. Among the compared methods, only GILT and OFA explicitly report pre-training across node-, link-, and graph-level tasks, while GraphAny and GCOPE focus only on node-level tasks and RiemannGFM covers node- and link-level tasks. This difference is important in the graph domain because our evaluation targets a unified graph in-context learning setting across multiple task granularities, so broader task diversity during pre-training is likely to improve robustness at test time.

Algorithm 1 Tuning-Free Inference of GILT

Require: Graph adjacency $\mathbf{A}$, node features $\mathbf{X}$, support set $\mathcal{S}$, query set $\mathcal{Q}$, target dimension d, GCN depth L_{gcn}, Transformer depth L_{tf}
Ensure: Prediction probability matrix $\mathbf{P}$ for the query set

if $\dim(\mathbf{X}) > d$ **then**
  $\mathbf{X}' \leftarrow \text{SVDProject}(\mathbf{X}, d)$
else
  $\mathbf{X}' \leftarrow \text{ZeroPad}(\text{SVDTransform}(\mathbf{X}), d)$
end if
$\mathbf{X}' \leftarrow \text{ColumnWiseStandardScale}(\mathbf{X}')$
$\tilde{\mathbf{A}} \leftarrow \text{NormalizeAdjacencyWithSelfLoops}(\mathbf{A})$
$\mathbf{H}^{(0)} \leftarrow \mathbf{X}'$
for $l = 1$ **to** L_{gcn} **do**
  $\mathbf{H}^{(l)} \leftarrow \text{LayerNorm}(\tilde{\mathbf{A}}\mathbf{H}^{(l-1)})$
end for
$\mathbf{H} \leftarrow \mathbf{H}^{(L_{gcn})}$
for each class $c \in \mathcal{C}$ **do**
  $\mathcal{S}_c \leftarrow \text{GetSamplesByClass}(\mathcal{S}, c)$
  $\mathbf{p}_c \leftarrow \ell_2\text{-Normalize}(\text{MeanPool}(\mathbf{H}[\mathcal{S}_c]))$
end for
$\mathbf{T}_{\mathcal{S}} \leftarrow \text{Concat}(\mathbf{H}[\mathcal{S}], \mathbf{p}_{\mathbf{Y}})$
$\mathbf{T}_{\mathcal{Q}} \leftarrow \text{Concat}(\mathbf{H}[\mathcal{Q}], \mathbf{0})$
for $l = 1$ **to** L_{tf} **do**
  $\mathbf{T}_{\mathcal{S}} \leftarrow \text{TransformerBlock}(\mathbf{T}_{\mathcal{S}}, \mathbf{T}_{\mathcal{S}}, \mathbf{T}_{\mathcal{S}})$
  $\mathbf{T}_{\mathcal{Q}} \leftarrow \text{TransformerBlock}(\mathbf{T}_{\mathcal{Q}}, \mathbf{T}_{\mathcal{S}}, \mathbf{T}_{\mathcal{S}})$
end for
$\mathbf{Z}_{\mathcal{S}} \leftarrow \text{ExtractClassSpace}(\mathbf{T}_{\mathcal{S}})$
$\mathbf{Z}_{\mathcal{Q}} \leftarrow \text{ExtractClassSpace}(\mathbf{T}_{\mathcal{Q}})$
for each class $c \in \mathcal{C}$ **do**
  $\mathbf{v}_c \leftarrow \ell_2\text{-Normalize}(\text{MeanPool}(\mathbf{Z}_{\mathcal{S}}[\mathcal{S}_c]))$
end for
for each query sample $j \in \mathcal{Q}$ **do**
  for each class $c \in \mathcal{C}$ **do**
    $\mathbf{Sim}[j, c] \leftarrow \langle \mathbf{Z}_{\mathcal{Q}}[j], \mathbf{v}_c \rangle$
  end for
  $\mathbf{P}[j, :] \leftarrow \text{Softmax}(\mathbf{Sim}[j, :])$
end for

return $\mathbf{P}$

Third, GILT also appears to cover a wider mix of application domains, including citation, social, transportation, web, and molecular graphs. In contrast, the reported pre-training domains of prior GFMs are more concentrated, with several methods focusing primarily on citation-style benchmarks and only limited extension to areas such as e-commerce, knowledge graphs, co-authorship, or molecular data. Taken together, these graph-domain comparisons suggest that GILT's stronger transferability is supported not only by larger pre-training scale, but also by a more diverse combination of tasks and domains relative to prior graph-domain GFMs.

E. Pseudo-code of Tuning-Free Inference

Algorithm 1 summarizes the inference pipeline of GILT, including feature pre-processing, graph-native structural encoding, task tokenization, in-context reasoning, and non-parametric prediction.

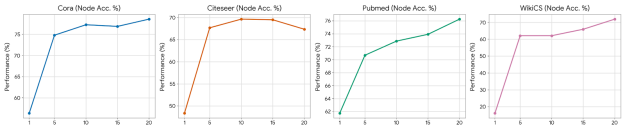

Figure 3. The influence of the number of shots (K) on GILT's few-shot performance. The x-axis represents the number of support examples per class, and the y-axis represents the classification accuracy on the test set. Each line corresponds to a different dataset.

F. Hyperparameter Settings

This section details the hyperparameter settings for GILT. To efficiently find a robust configuration, we performed a hyperparameter search using Bayesian optimization. The search was conducted on a held-out set of validation tasks sampled from our pre-training datasets to identify a single, robust set of parameters. The final optimal values, listed in Table 10, were then frozen and used for all reported experiments across all datasets and tasks.

G. Investigation into the Influence of Shot Number

To better understand how GILT utilizes contextual information, we analyzed its performance across a varying number of support examples (shots) for the node classification task. The results are presented in Figure 3.

As illustrated in the figure, there is a clear positive correlation between the number of shots provided in the context and the model's overall performance. The most significant gains in accuracy are typically observed when increasing the shot number from one to ten. As more examples are added to the context, the performance continues to improve, though the rate of improvement gradually diminishes. This analysis confirms that providing a richer context with more examples is an effective method for enhancing GILT's predictive accuracy, which is consistent with the expected behavior of an in-context learning framework.

H. LLM Usage

We utilized LLMs to assist in the preparation of this work. Specifically, we used LLMs for debugging code snippets and for proofreading and improving the clarity of the manuscript's text. The authors reviewed and edited all LLM-generated content and take full responsibility for the final submission.

Table 10. Hyperparameter search space and optimal values used for the GILT model.

Hyperparameter	Search Space	Optimal Value
Pre-training Hyperparameters		
Learning Rate	[1e-6, 1e-3]	1e-5
Weight Decay	[1e-6, 1e-2]	1e-4
Model Dropout	[0, 0.5]	0.2
Total Training Epochs	10, 50, 100	50
Multi-Task Training Details		
Batch Size (Node Tasks)	{2048, 4096, 8192, 16384}	2048
Batch Size (Link Tasks)	{4096, 8192, 16384, 32768}	16384
Batch Size (Graph Tasks)	{1024, 2048, 4096}	1024
GILT Architecture Hyperparameters		
Unified Dimension (d)	64, 128, 256	128
GCN Encoder Layers	[2, 8]	6
Transformer Layers (L)	[1, 5]	2