UniGTE: Unified Graph–Text Encoding for Zero-Shot Generalization across Graph Tasks and Domains

Duo Wang **Yuan Zuo*** **Guangyue Lu** **Junjie Wu**
MIIT Key Laboratory of Data Intelligence and Management, Beihang University
{wangduo58, zuoyuan, lugybuaa, wujj}@buaa.edu.cn

Abstract

Generalizing to unseen graph tasks without task-specific supervision is challenging: conventional graph neural networks are typically tied to a fixed label space, while large language models (LLMs) struggle to capture graph structure. We introduce **UniGTE**, an instruction-tuned encoder–decoder framework that unifies structural and semantic reasoning. The encoder augments a pretrained autoregressive LLM with learnable *alignment tokens* and a *structure-aware graph–text attention* mechanism, enabling it to attend jointly to a tokenized graph and a natural-language task prompt while remaining permutation-invariant to node order. This yields compact, *task-aware graph representations*. Conditioned solely on these representations, a frozen LLM decoder *predicts and reconstructs*: it outputs the task answer and simultaneously paraphrases the input graph in natural language. The reconstruction objective regularizes the encoder to preserve structural cues. UniGTE is instruction-tuned on five datasets spanning node-, edge-, and graph-level tasks across diverse domains, yet requires no fine-tuning at inference. It achieves new state-of-the-art zero-shot results on node classification, link prediction, graph classification and graph regression under cross-task and cross-domain settings, demonstrating that tight integration of graph structure with LLM semantics enables robust, transferable graph reasoning.

1 Introduction

Zero-shot learning in graph machine learning seeks to generalize to unseen tasks and domains without task-specific supervision. Although graph neural networks (GNNs) excel in fully supervised settings, they transfer poorly to new label spaces or data distributions without costly fine-tuning [1]. Inspired by recent progress in natural language processing (NLP), prompt-based extensions have been proposed to enhance GNN generalization [2, 3]. However, the rigid architecture of conventional GNNs—especially their task-specific output heads—still hampers adaptability in zero-shot scenarios.

The advent of large language models (LLMs) opens new avenues for zero-shot reasoning on graphs. A direct approach serializes graph data into text and feeds it to an LLM [4, 5, 6, 7]. While simple, this often underperforms because LLMs lack structural inductive bias [8]. Recent work therefore explores combining GNNs and LLMs, which can be grouped as follows.

LLMs as enhancers. These methods keep a GNN as the primary predictor and employ the LLM only to inject auxiliary semantic signals—for example, generating synthetic labels or textual node descriptions [9, 10, 11, 12]. Although such signals improve performance, the approaches inherit the architectural rigidity of GNNs and typically require retraining for new tasks. Replacing task-specific output heads with textual label embeddings enables limited zero-shot classification [13, 14] but does not naturally extend to regression or other objectives, and semantic mismatch between graph and text remains an issue.

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

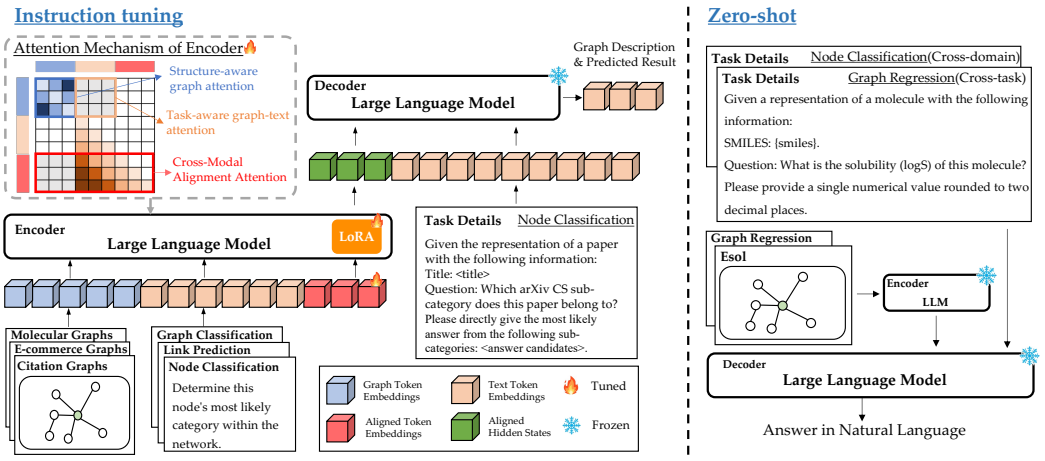


Figure 1: Framework of UniGTE

LLMs as predictors. Here the predictive role is assigned to the LLM, while a GNN supplies structural information aligned to the LLM's semantic space—usually via self-supervised pretraining and cross-modal projection [15, 16, 17]. Because the two components are trained separately, it is hard to inject task-specific signals in a task-aware manner, limiting generalization. Deeper integrations such as GOFA [18] inject GNN features into LLM tokens at inference time, boosting zero-shot performance but at a high computational cost and with persistent cross-task and cross-domain gaps.

Our proposal. We introduce **UniGTE**, a unified encoder–decoder framework that is instruction-tuned on a diverse suite of graph datasets (node classification, link prediction, graph classification; domains include citation networks, e-commerce graphs, molecular structures). The *encoder* augments a pretrained autoregressive LLM to jointly consume a tokenized graph, a natural-language task prompt, and a fixed set of learnable *alignment tokens*. During self-attention, these alignment tokens aggregate structural and prompt signals into a compact *task-aware graph representation*. The *decoder* is a frozen autoregressive LLM that conditions on this representation to (i) generate the task prediction and (ii) reconstruct the graph prompt, with the latter providing auxiliary supervision via a prompt-level loss. This design yields a single model that is permutation-invariant to node order, conditioned on task instructions, and capable of zero-shot generalization across modalities, tasks, and domains. Our key contributions are as follows:

- We present UniGTE—the first unified encoder–decoder architecture that achieves zero-shot generalization across diverse graph tasks and domains without any task-specific fine-tuning.
- UniGTE conditions graph representation learning on task prompts and embeds both graph structure and textual semantics in a common space, enabling flexible adaptation across modalities and objectives.
- Extensive experiments demonstrate state-of-the-art zero-shot results on node classification, link prediction, and graph regression across multiple domains.

2 Methodology

We present **UniGTE**, a unified encoder–decoder framework for learning transferable graph representations across heterogeneous tasks and domains. UniGTE is instruction-tuned on a diverse collection of graph datasets that cover multiple task families—node classification, link prediction, and graph classification—and domains such as citation networks, e-commerce graphs, and molecular structures.

The architecture comprises an encoder and a decoder. The *encoder* extends a pretrained autoregressive large language model (LLM) to jointly process graph-structured inputs and natural-language task prompts. Its input sequence comprises a tokenized graph (e.g., node representations from a subgraph), a task-specific prompt, and a fixed set of learnable *alignment tokens*. These alignment tokens act as cross-modal anchors during self-attention, aggregating information from graph tokens guided by the

task prompt and distilling it into a compact, task-aware latent representation—termed the *task-aware graph representation*.

The *decoder* is a frozen autoregressive LLM that conditions exclusively on the encoder outputs of the alignment tokens—that is, on the task-aware graph representation. It autoregressively generates two outputs: (i) the task prediction and (ii) a reconstruction of the graph prompt. The latter serves as an auxiliary supervision signal, implemented via a prompt-level loss that encourages the encoder to preserve the semantic content of the input graph. An overview of the framework is shown in Fig. 1.

2.1 Task definition and notation

A graph-learning task $\tau \in \mathcal{T}$ is formally defined as $\tau := \{(\mathcal{G}_i, y_i^\tau)\}_{i=1}^M \cup \{T^\tau\}$, where M is the number of graph instances. Each instance pairs a graph $\mathcal{G}_i$ with a target output y_i^τ, expressed as a sequence of text tokens that typically encodes a class label or a numerical value.

A graph is denoted $\mathcal{G} = \big(\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{X}, \mathbf{E}\big)$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the node set and $\mathcal{E} = \{e_1, e_2, \ldots, e_{|\mathcal{E}|}\}$ the edge set. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ is defined by $\mathbf{A}_{ij} = 1$ iff $(v_i, v_j) \in \mathcal{E}$. Node features are stored in $\mathbf{X} \in \mathbb{R}^{N\times F_N}$ and edge features in $\mathbf{E} \in \mathbb{R}^{|\mathcal{E}|\times F_E}$, with F_N and F_E denoting their respective feature dimensions.

Each task τ is accompanied by a natural-language instruction $T^\tau = [T^\tau_{\text{desc}}, T^\tau_{\text{detail}}]$, containing two components. The task description $T^\tau_{\text{desc}} = [w_1^{(d)}, \ldots, w_\ell^{(d)}]$ briefly states the objective—for example, *"Determine this node's most likely category within the network's classification schema."* The task-specific input $T^\tau_{\text{detail}} = [w_1^{(s)}, \ldots, w_r^{(s)}]$ provides the contextualised information needed to solve the task, such as *"Given a representation of a paper with the following information: Title: {title}, Abstract: {abstract}. Question: Which arXiv CS sub-category does this paper belong to?"*

2.2 Unified graph-text encoder: learning task-aware graph representations via LLM

In UniGTE, the encoder receives a graph instance $\mathcal{G}_i$, the task description T^τ_{desc}, and a fixed set of alignment tokens $\mathcal{A}$ (defined below), and embeds them into a shared latent space. The decoder, conditioned on the encoder outputs corresponding to the alignment tokens—i.e., the task-aware graph representation—and on the task-specific input T^τ_{detail}, then autoregressively produces the target sequence y_i^τ.

We first describe the unified graph–text encoder. Prior work shows that graph tasks depend on structural and attribute information to different extents [19, 20]. To satisfy these varied requirements, the encoder must inject task cues into the aggregation process while retaining the ability to generalise across tasks and domains for zero-shot transfer. Inspired by the strong generalisation capacity of large language models (LLMs) in graph contexts, we build the encoder on a pretrained LLM that jointly encodes graph structure and task instructions, thereby learning task-aware representations.

2.2.1 Unified input formatting for graph tasks

To enable joint training across heterogeneous tasks and domains, we cast every task into a unified *graph-level* input. For node- and edge-level tasks, we extract an n-hop subgraph centred on the target node or edge and treat all nodes in this subgraph as input tokens $T_{\mathcal{G}}$:

$$T_{\mathcal{G}} = [w_1^{(g)}, \ldots, w_n^{(g)}],$$

where each token $w_i^{(g)}$ is obtained by encoding the node's attribute text with a pretrained language model (PLM), and $n = |\mathcal{G}|$ is the number of nodes in the subgraph. This abstraction lifts instance-level tasks to the graph level, promoting parameter sharing across task types [3].

To inject task semantics, we employ a unified description template T_{desc} that specifies both the task type and a concise summary of the input graph. Presenting this textual prompt alongside the graph tokens guides the encoder to produce a latent representation that is simultaneously structure- and task-aware.

To bridge graph and text inputs in a shared semantic space, we append a fixed set of learnable *alignment tokens*:

$$\mathcal{A} = [\,\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m\,], \qquad m \ll n + \ell,$$

where m is much smaller than the combined number of graph tokens n and text tokens ℓ. These tokens distil and integrate information from both modalities and act as the *sole* interface to the decoder. This design unifies the representational spaces of graph and text while shrinking the overall token budget, allowing the model to accommodate longer inputs within its sequence-length limit.

Given $T_{\mathcal{G}}$, T_{desc}, and $\mathcal{A}$, the encoder input is

$$\mathbf{x}_{\text{enc}} = \left[T_{\mathcal{G}} \;;\; T_{\text{desc}} \;;\; \mathcal{A} \right] \in \mathbb{R}^{(n+\ell+m)\times d_h},$$

where d_h is the hidden dimension of the underlying language model.

2.2.2 Structure-aware graph-text attention

Beyond simple input formatting, we propose a *structure-aware graph–text attention* mechanism. Existing methods often linearize graph nodes into token sequences and input them into large language models (LLMs). However, their performance heavily relies on a *fixed* node ordering and degrades significantly under permutation [21]. This fragility stems from the absolute positional encoding in standard self-attention, which contradicts the permutation invariance inherent to graph data. To address this, we design a unified attention mechanism that jointly processes graph and text tokens while preserving structural invariance.

Let the projection matrices be $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d_h \times d_k}$ and $\mathbf{W}_O \in \mathbb{R}^{d_k \times d_h}$. The scaled dot-product attention with *relative* position encoding, as applied in LLMs, is defined as

$$\hat{\mathbf{A}}_{ij} = \frac{(\mathbf{x}_i \mathbf{W}_Q)\, \mathbf{R}(i-j)\, (\mathbf{x}_j \mathbf{W}_K)^\top}{\sqrt{d_k}}, \quad \mathbf{A} = \mathrm{softmax}(\hat{\mathbf{A}}), \quad \mathrm{Attn}(\mathbf{X}) = \mathbf{A}(\mathbf{X}\mathbf{W}_V)\mathbf{W}_O, \quad (1)$$

where $\mathbf{R}(\cdot)$ denotes the rotary positional encoding (RoPE) transformation [22].

Cross-modal RoPE We extend RoPE to accommodate graph, text, and alignment tokens. For *text* tokens, i and j represent absolute positions, and $\mathbf{R}(i-j)$ operates as in standard RoPE. For *graph* tokens, we assign a shared, learnable position index `<GraphPos>`, such that $i - j = 0$ for all graph-node pairs. This causes $\mathbf{R}(0)$ to reduce to the identity matrix, removing order sensitivity. For cross-modal interactions, we compute the offset between text token positions and `<GraphPos>`, enabling the model to learn consistent alignments across modalities.

While this construction restores node-permutation invariance, it discards explicit structural cues. We re-inject them through additive biases:

$$\hat{\mathbf{A}}_{ij} = \frac{(\mathbf{x}_i \mathbf{W}_Q)\, \mathbf{R}(i-j)\, (\mathbf{x}_j \mathbf{W}_K)^\top}{\sqrt{d_k}} \;+\; b(i,j), \quad (2)$$

with $b(i,j) = \mathbb{I}_{\{i,j\le n\}} \big(b^{\text{PE}}_{ij} + b^{\text{Edge}}_{ij}\big) + b^{\text{M}}_{ij}$. The three bias terms are:

Distance-based structural bias Following graph transformers [23], we embed the shortest-path distance between graph nodes as:

$$b^{\text{PE}}_{ij} = e\big(\mathrm{dist}_{\mathcal{G}}(i,j)\big), \quad (3)$$

where $e(\cdot)$ is a learnable lookup table, and $\mathrm{dist}_{\mathcal{G}}(i,j)$ denotes the shortest-path distance between nodes i and j.

Edge-aware bias Graphs in practice feature heterogeneous edge types. For each edge e_k on the shortest path $\mathrm{SP}(i,j)$ we compute

$$b_{e_k} = \mathrm{MLP}\Big(\mathrm{PLM}\big(\text{“Description of } e_k\text{”}\big)\Big) \quad (4)$$

and aggregate by $b^{\text{Edge}}_{ij} = \frac{1}{|\,\mathrm{SP}(i,j)|} \sum_{k\in \mathrm{SP}(i,j)} b_{e_k}$. Unlike discrete type embeddings [24], this natural-language description lets the model generalize to unseen edge semantics, benefiting zero-shot transfer.

Masking bias Finally, b^{M}_{ij} enforces directional constraints:

$$b^{\mathrm{M}}_{ij} = \begin{cases} -\infty, & \text{if } (i > j \ \wedge \ i > n) \text{ or } (i \le n \ \wedge \ n < j \le n + \ell), \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Graph tokens attend bidirectionally to capture connectivity; in addition, they may look *forward* into the text so downstream instructions can refine their representations. Conversely, text tokens never attend to graph tokens, preventing information leakage. Alignment tokens obey the standard causal mask across both modalities.

Together, these biases endow our attention with (i) permutation invariance for graph inputs, (ii) explicit structural awareness, and (iii) flexible cross-modal interaction, enabling robust reasoning across diverse graph-text tasks.

2.2.3 Instruction tuning of the unified graph–text encoder

After the encoder processes the input graph and prompt, we take the hidden states $\mathcal{H}_{\mathcal{A}}$ at the special *alignment tokens* $\mathcal{A}$ as task-aware graph representations. $\mathcal{H}_{\mathcal{A}}$ fuse structural cues from the graph with semantic hints from the task description and constitute the sole conditioning signal for the decoder.

Instruction-tuning objective. We train concurrently on node-, edge- and graph-level tasks drawn from multiple domains. For a task τ with target sequence $\boldsymbol{y}^{\tau} = (y^{\tau}_1, \ldots, y^{\tau}_{L_\tau})$ and instance-specific instruction T^{τ}_{detail}, the negative log-likelihood loss is

$$\mathcal{L}^{\tau}_{\text{IT}}(\theta_{\text{enc}}) \ = \ -\sum_{t=1}^{L_\tau} \log P\big(y^{\tau}_t \mid \mathcal{H}_{\mathcal{A}},\, T^{\tau}_{\text{detail}},\, y^{\tau}_{<t};\, \theta_{\text{enc}}\big). \tag{6}$$

Here T^{τ}_{detail} differs from the terse task description T^{τ}_{desc} in that it contains the concrete, instance-level input needed to solve τ.

During instruction tuning we keep the decoder frozen and update only a small subset of encoder parameters: LoRA adapters θ_{LoRA}, alignment-token embeddings $\theta_{\mathcal{A}}$, the MLP weights θ_{MLP} that compute edge-aware bias, and the table θ_{e} for relative-position bias. We denote their union by θ_{enc}.

Auxiliary prompt reconstruction. We additionally ask the decoder to reconstruct the graph description $d_{\mathcal{G}}$ embedded in the prompt. This supervision encourages the alignment tokens to better encode structural information, while eliminating the need for a separate autoencoding stage. With target tokens $\boldsymbol{w}^{(d_{\mathcal{G}})} = (w^{(d_{\mathcal{G}})}_1, \ldots, w^{(d_{\mathcal{G}})}_{L_d})$, the auxiliary loss is

$$\mathcal{L}^{\tau}_{\text{prompt}}(\theta_{\text{enc}}) \ = \ -\sum_{t=1}^{L_d} \log P\big(w^{(d_{\mathcal{G}})}_t \mid \mathcal{H}_{\mathcal{A}},\, w^{(d_{\mathcal{G}})}_{<t};\, \theta_{\text{enc}}\big). \tag{7}$$

Overall objective. The total loss for task τ is the sum of the two terms:

$$\mathcal{L}^{\tau}_{\text{total}}(\theta_{\text{enc}}) \ = \ \mathcal{L}^{\tau}_{\text{IT}}(\theta_{\text{enc}}) \ + \ \mathcal{L}^{\tau}_{\text{prompt}}(\theta_{\text{enc}}). \tag{8}$$

In training we minimise $\sum_{\tau} \mathcal{L}^{\tau}_{\text{total}}(\theta_{\text{enc}})$ with respect to θ_{enc}.

2.3 Training and evaluation strategy

To assess the scalability and generalization ability of our model, we curate a diverse set of graph datasets spanning multiple levels (node, edge, and graph) and domains. The benchmark includes both classification and regression tasks drawn from application areas such as citation networks, e-commerce platforms, social media, and molecular graphs. Specifically, it comprises 17 datasets from five distinct domains, covering node classification, link prediction, graph classification, and graph regression tasks. Full details of the datasets are provided in the Appendix A. We use a subset of these datasets for instruction tuning, and directly evaluate the model's zero-shot performance on the remaining datasets without any further fine-tuning.

3 Experimental results

In this section, we conduct comprehensive experiments to validate the effectiveness of UniGTE. Our evaluation is designed to address the following research questions:

RQ1: How well does UniGTE generalize to unseen datasets within the same domain (in-domain zero-shot)?

RQ2: Can UniGTE handle more challenging generalization settings, such as transferring across domains or tasks unseen during training (cross-domain and cross-task zero-shot)?

RQ3: What are the respective contributions of task-aware graph encoding and alignment tokens to the zero-shot performance of UniGTE?

3.1 Experimental setup

Datasets We jointly train UniGTE on five datasets: Arxiv [25], Children [26], Computer [26], FB15K237 [13], and ChEMBL [27], spanning node classification, link prediction, and graph classification tasks. For Arxiv, Children, and Computer, we construct both node classification and link prediction tasks to increase task diversity. After training, we evaluate UniGTE in a zero-shot setting on a set of unseen datasets. For in-domain evaluation, we use datasets from the same domains (e.g., additional citation or e-commerce graphs). To assess cross-domain generalization, we evaluate on datasets from different domains such as web graphs and social networks. Finally, to evaluate cross-task generalization, we include a previously unseen graph regression task. Detailed descriptions of all training and evaluation datasets are provided in the Appendix A.

Baselines We compare UniGTE against several recent state-of-the-art models with demonstrated transfer and zero-shot capabilities. **OFA** [13] combines a GNN-based predictor with a large language model (LLM) via prompt-based input augmentation. **GraphGPT** [15], **LLaGA** [28], and **TEA-GLM** [17] adopt an LLM as the primary predictor and align graph-text representations through either a multi-layer perceptron (MLP) or a linear projection layer. **GOFA** [18] also employs an LLM as the predictor but incorporates structural information through inter-layer graph aggregation within the LLM architecture. Due to the high computational cost of training and evaluating **GraphGPT**, we report its results as provided in the original publication. For all other baselines, we re-ran the official implementations and conducted evaluations under our experimental setup. Detailed settings of the experimental environment can be found in Appendix C.

3.2 In-domain zero-shot generalization (RQ1)

To address RQ1, we evaluate each model's zero-shot performance on datasets from the same domains as those seen during training. These include citation networks (**Pubmed** [29] and **Cora** [30], with the latter being a more challenging variant featuring 70 classes), e-commerce datasets (**Photo** and **Sports**), and molecular graphs [31](**HIV**, **BACE**, and **PCBA**). We report accuracy for node classification, and AUC for link prediction and graph classification, reflecting the standard metrics used for each task type.

As shown in Table 1, UniGTE achieves the best overall performance across tasks and datasets, outperforming all baselines on the majority of benchmarks. Models relying on GNN-based predictors, such as OFA, struggle to generalize and exhibit weak transfer performance. Surprisingly, in most tasks, LLM-based methods like LLaGA and GraphGPT fail to outperform their base model, Vicuna-7B, suggesting that their lack of permutation invariance hinders generalization—changes in node ordering significantly impact their predictions.

Among the baselines, TEA-GLM applies a pooling mechanism to produce a fixed number of graph tokens, which preserves permutation invariance and contributes to better generalization in node classification and link prediction tasks. However, its inability to incorporate task-specific signals leads to inconsistent performance across tasks and even negative transfer in graph classification. GOFA, trained using our instruction tuning pipeline on the official pre-trained checkpoint, achieves limited gains. Despite extensive pretraining, it underperforms in most tasks and fails to match even an untuned LLM in many cases.

In contrast, UniGTE demonstrates consistent positive transfer across all datasets and task types. This can be attributed to its use of *task-specific signals during graph encoding*, which enable the

Table 1: In-domain zero-shot results. **Bold** and underline indicate the best and second-best results, respectively. N.S. denotes unsupported tasks.

Model	Pubmed	Cora	Photo	Sports	BACE	HIV	PCBA	Pubmed	Photo
	Node Classification				Graph Classification			Link Prediction	
Vicuna-7B	0.721	0.155	0.384	0.371	0.492	0.467	0.497	0.502	0.576
OFA	0.237	0.189	0.317	0.047	0.483	0.404	0.424	0.499	0.499
GraphGPT-std	0.701	0.126	–	–	–	–	–	0.501	–
LLaGA	0.726	0.156	0.249	0.351	N.S.	N.S.	N.S.	**0.740**	0.659
TEA-GLM	0.781	0.202	0.418	0.357	0.467	0.498	0.434	0.663	0.675
GOFA	0.614	0.039	0.447	0.133	0.500	0.481	0.500	0.507	0.504
UniGTE	**0.870**	**0.215**	**0.565**	**0.403**	**0.534**	**0.501**	**0.541**	0.722	**0.732**

model to distinguish between task objectives, and its *alignment tokens*, which unify structural and semantic information. Together, these components contribute to UniGTE's superior generalization in in-domain, zero-shot scenarios.

3.3 Cross-domain and cross-task generalization (RQ2)

To evaluate the generalization ability of each model in more challenging settings, we conduct zero-shot testing under both *cross-domain* and *cross-task* conditions. Specifically, we use datasets from domains not seen during training—**WikiCS** [32] (web links), **Reddit**, and **Instagram** [33](social networks)—as well as an entirely new task: *graph regression*, evaluated on **Esol** [34], **Lipo**, and **Freesolv** [35]. We report accuracy for node classification and mean absolute error (MAE) for regression.

As shown in Table 2, UniGTE outperforms all baselines across both domains and task types. Most baseline models show some degree of positive transfer, but results remain inconsistent. LLaGA and OFA do not support graph-level tasks due to limitations in model design. GOFA achieves stronger regression performance than TEA-GLM on some datasets, yet both models show clear trade-offs: they perform well on specific domains or tasks but fail to generalize broadly.

In contrast, UniGTE consistently delivers strong performance across all datasets and settings. The challenging nature of cross-domain and cross-task transfer underscores the importance of robust generalization. UniGTE's results demonstrate its ability to generalize effectively beyond the training distribution, highlighting the benefits of its unified graph-text representation and task-aware alignment mechanism.

3.4 Ablation study (RQ3)

We conduct ablation studies to assess the contributions of two key components in UniGTE: **alignment tokens** and **task-aware graph encoding**. To evaluate the role of alignment tokens, we remove them entirely and allow the decoder to generate outputs without their guidance. For task-aware graph encoding, we replace the task-specific description T_{desc} with a fixed, generic prompt that is not tailored to any specific task.

To provide a comprehensive analysis, we evaluate performance from both **domain-level** and **task-level** perspectives, averaging metrics across datasets within each category. To ensure comparability across different task types—particularly between classification and regression—we adopt a normalized MAE score defined as:

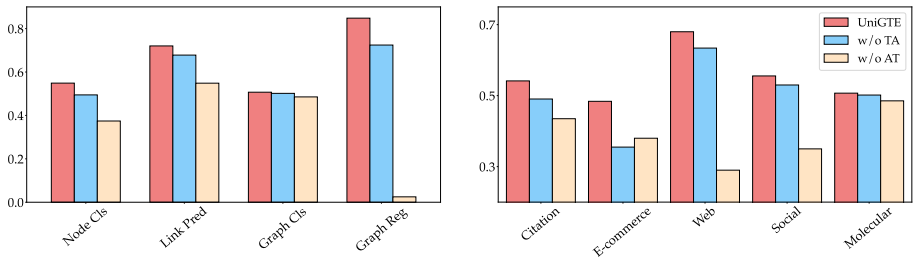
$$\widehat{\text{MAE}} = 1 - \frac{\text{MAE} - \text{MAE}_{\min}}{\text{MAE}_{\max} - \text{MAE}_{\min}},$$

where $\text{MAE}_{\min}$ and $\text{MAE}_{\max}$ denote the minimum and maximum MAE values observed across all models and datasets. This normalization yields scores between 0 and 1, where higher values indicate better performance, thus making them directly comparable to metrics such as accuracy and AUC.

As shown in Figure 2, both components contribute substantially to the model's generalization. "**w/o AT**" corresponds to the setting without alignment tokens, and "**w/o TA**" reflects the ablation of task-aware graph encoding. Across both perspectives, removing either component results in a consistent performance drop. The absence of alignment tokens significantly degrades performance, highlighting their role in capturing structured and semantic information essential for zero-shot inference. Similarly,

Table 2: Zero-shot performance on cross-domain node classification and cross-task graph regression. **Bold** and underline indicate the best and second-best results, respectively. N.S. indicates tasks not supported by the model. Lower is better for regression (MAE).

Model	WikiCS	Reddit	Instagram	Esol	Lipo	Freesolv
	Node Classification (Accuracy)			Graph Regression (MAE)		
Vicuna-7B	0.290	0.309	0.391	6.58	11.22	64.11
OFA	0.361	0.498	0.580	N.S.	N.S.	N.S.
LLaGA	0.601	0.499	0.397	N.S.	N.S.	N.S.
TEA-GLM	0.449	0.491	0.479	14.90	9.76	13.35
GOFA	0.613	0.493	0.367	4.93	1.36	14.98
UniGTE	**0.680**	**0.510**	**0.601**	**2.54**	**1.03**	**9.18**



(a) Domain-level perspective (b) Task-level perspective

Figure 2: Ablation study of alignment tokens and task-aware graph encoding, reported from both domain-level and task-level perspectives. "w/o AT" indicates the removal of alignment tokens, while "w/o TA" refers to replacing task-specific descriptions with a generic prompt.

removing task-specific descriptions reduces performance across the board, confirming the importance of task-aware encoding in providing fine-grained task signals.

Overall, these results underscore the necessity of both components: alignment tokens enhance generalization by bridging modalities, while task-aware encoding improves adaptability to diverse task objectives.

4 Related work

In this section, we review recent advances in zero-shot and transfer learning for graph machine learning. Existing approaches can be broadly categorized into two groups: (1) methods based solely on GNNs, and (2) methods that incorporate large language models (LLMs). The LLM-based methods can be further divided into two subtypes: *LLM as an enhancer*, where the LLM provides auxiliary semantic information to assist graph models, and *LLM as a predictor*, where the LLM itself performs prediction after being aligned with structural features.

4.1 Zero-shot and transfer learning with GNNs

A wide range of self-supervised learning techniques have been proposed to reduce the reliance of graph neural networks (GNNs) on labeled data and improve their generalization. For example, Deep Graph Infomax (DGI) [36] maximizes mutual information between local and global embeddings. Contrastive methods such as GraphCL [37], GCA [38], GCC [39], and JOAO [40] construct positive and negative views of graphs to learn invariant node representations. Generative approaches like GraphMAE [41, 42] adopt masked reconstruction objectives to jointly encode semantic and structural cues.

Although effective, these models typically require task-specific fine-tuning on downstream datasets. To improve transferability, prompt-based GNNs have recently gained attention. GraphPrompt [2] proposes a unified prompt template shared across pretraining and fine-tuning, improving knowledge transfer. ProG [3] reformulates node-level, link-level, and graph-level tasks into a unified prompting

framework and leverages meta-learning for multi-task adaptation. However, these approaches still depend heavily on GNN-specific architectures, making it difficult to generalize across tasks or datasets with varying output spaces without further retraining.

4.2 LLMs for graph zero-shot and transfer learning

4.2.1 LLMs as enhancers

One line of work leverages LLMs as *semantic enhancers*—generating task descriptions, pseudo-labels, or contextual cues to guide GNNs during training [10, 11, 12, 13, 14]. These methods benefit from the rich prior knowledge encoded in LLMs; however, the final prediction is still performed by GNNs, limiting their flexibility and generalization in unseen tasks. Some studies [13, 14] further propose novel architectural designs that enable GNNs to support zero-shot transfer across datasets. Despite these improvements, such approaches still exhibit limited generalization ability and are inherently unsuitable for regression tasks.

4.2.2 LLMs as predictors

A second line of work treats LLMs as *predictors*. Some studies attempt to serialize graph data into textual sequences [5, 6, 4], which are then directly fed into LLMs for inference. While this approach enables zero-shot reasoning by leveraging pretrained language models, it often compromises structural fidelity, as LLMs primarily capture token-level co-occurrence patterns rather than graph-specific inductive biases [8].

Some methods like LLaGA [28] attempt to convert nodes into token sequences and input them directly into LLMs. However, this approach loses permutation invariance—a key property of graph data—making the output highly sensitive to node ordering [21]. To better preserve structural information, other works incorporate GNNs to extract graph features, which are then aligned with LLM inputs. For example, GraphGPT [15] aligns a graph transformer encoder with an LLM through a two-stage training process. UniGraph [16] employs masked word prediction to pretrain a GNN and aligns it with LLM embeddings via an MLP-based projection. TEA-GLM [17] further proposes a feature-wise contrastive pretraining strategy, followed by lightweight projection for alignment.vDespite these advances, most existing methods treat GNNs and LLMs as loosely coupled modules and perform alignment in a post hoc manner, limiting their adaptability to task-specific signals. As a result, such models often struggle in multi-task or cross-domain scenarios.

GOFA [18] addresses this limitation by introducing a tighter integration strategy, inserting GNN layers between LLM transformer blocks to enable token-level structural aggregation. However, this architecture incurs high computational costs and still faces performance challenges in zero-shot settings across diverse tasks and domains.

5 Limitations

While UniGTE achieves strong zero-shot generalization across a wide range of graph tasks and domains, its performance gains on link prediction tasks are less pronounced compared to node classification and graph regression. This may be due to the pairwise nature of link prediction, which poses unique challenges for prompt formulation and representation alignment. Exploring more effective strategies for encoding edge-level interactions and adapting task prompts for link prediction remains an important direction for future work.

6 Conclusion

We presented **UniGTE**, a unified encoder–decoder framework for zero-shot graph learning. UniGTE fuses graph structure and natural-language task instructions in a shared representation space via a permutation-invariant encoder with learnable alignment tokens, while a frozen LLM decoder produces both task predictions and prompt reconstructions. This design supports flexible transfer across node-, edge-, and graph-level tasks and across disparate domains. Extensive experiments verify UniGTE's robustness, setting new zero-shot state-of-the-art results under demanding cross-task and cross-domain conditions. Our study highlights the benefit of tightly integrating structural and semantic cues for broadly transferable graph reasoning.

7 Acknowledgement

This work was supported by the National Key R&D Program of China (2023YFC3304700). The work of Yuan Zuo was partially supported by the National Natural Science Foundation of China (72571019, 72531002) and the Shenzhen Science and Technology Program (CJGJZD20230724093201004). Dr. Junjie Wu's work was partially supported by the National Natural Science Foundation of China (72031001, 72242101) and Outstanding Young Scientist Program of Beijing Universities (JWZQ20240201002).

References

[1] Mingxuan Ju, Tong Zhao, Qianlong Wen, Wenhao Yu, Neil Shah, Yanfang Ye, and Chuxu Zhang. Multi-task self-supervised graph neural networks enable stronger task generalization. In *The Eleventh International Conference on Learning Representations*, 2023.

[2] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM Web Conference 2023*, page 417–428, 2023.

[3] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. All in one: Multi-task prompting for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 2120–2131, 2023.

[4] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. Exploring the potential of large language models (LLMs) in learning on graph. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*, 2023.

[5] Jiayan Guo, Lun Du, and Hengyu Liu. Gpt4graph: Can large language models understand graph structured data ? an empirical evaluation and benchmarking. *ArXiv*, abs/2305.15066, 2023.

[6] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can language models solve graph problems in natural language? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[7] Chang Liu and Bo Wu. Evaluating large language models on graphs: Performance insights and comparative analysis. *arXiv preprint arXiv:2308.11224*, 2023.

[8] Jin Huang, Xingjian Zhang, Qiaozhu Mei, and Jiaqi Ma. Can llms effectively leverage graph structural information: when and why. *arXiv preprint arXiv:2309.16595*, 2023.

[9] Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. Natural language is all a graph needs. *arXiv preprint arXiv:2308.07134*, 2023.

[10] Jianxiang Yu, Yuxiang Ren, Chenghua Gong, Jiaqi Tan, Xiang Li, and Xuecang Zhang. Empower text-attributed graphs learning with large language models (llms). *arXiv preprint arXiv:2310.09872*, 2023.

[11] Lianghao Xia, Ben Kao, and Chao Huang. Opengraph: Towards open graph foundation models. *arXiv preprint arXiv:2403.01121*, 2024.

[12] Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. Label-free node classification on graphs with large language models (LLMs). In *The Twelfth International Conference on Learning Representations*, 2024.

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

[14] Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. Zerog: Investigating cross-dataset zero-shot transferability in graphs. *arXiv preprint arXiv:2402.11235*, 2024.

[15] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. *arXiv preprint arXiv:2310.13023*, 2023.

[16] Yufei He and Bryan Hooi. Unigraph: Learning a cross-domain graph foundation model from natural language. *arXiv preprint arXiv:2402.13630*, 2024.

[17] Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. Llms as zero-shot graph learners: Alignment of gnn representations with llm token embeddings. In *Advances in Neural Information Processing Systems*, 2024.

[18] Lecheng Kong, Jiarui Feng, Hao Liu, Chengsong Huang, Jiaxin Huang, Yixin Chen, and Muhan Zhang. GOFA: A generative one-for-all model for joint graph language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025.

[19] Duong Chi Thang, Hoang Thanh Dat, Nguyen Thanh Tam, Jun Jo, Nguyen Quoc Viet Hung, and Karl Aberer. Nature vs. nurture: Feature vs. structure for graph neural networks. *Pattern Recognition Letters*, 2022.

[20] Diana Gomes, Frederik Ruelens, Kyriakos Efthymiadis, Ann Nowe, and Peter Vrancx. When are graph neural networks better than structure-agnostic methods? In *I Can't Believe It's Not Better Workshop: Understanding Deep Learning Through Empirical Falsification*, 2022.

[21] Xixi Wu, Yifei Shen, Caihua Shan, Kaitao Song, Siwei Wang, Bohang Zhang, Jiarui Feng, Hong Cheng, Wei Chen, Yun Xiong, and Dongsheng Li. Can graph learning improve planning in LLM-based agents? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[22] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[23] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. In *Advances in Neural Information Processing Systems*, 2019.

[24] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, pages 28877–28888, 2021.

[25] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, pages 22118–22133, 2020.

[26] Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, Weiwei Deng, Qi Zhang, Lichao Sun, Xing Xie, and Senzhang Wang. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[27] Anna Gaulton, Louisa J. Bellis, A. Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and John P. Overington. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 2011.

[28] Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. Llaga: Large language and graph assistant. In *ICML*, 2024.

[29] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: LLM-to-LM interpreter for enhanced text-attributed graph representation learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[30] Zhihao Wen and Yuan Fang. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 506–516, 2023.

[31] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: A benchmark for molecular machine learning, 2018.

[32] Péter Mernyei and Cătălina Cangea. Wiki-cs: A wikipedia-based benchmark for graph neural networks, 2022.

[33] Yuhan Li, Peisong Wang, Xiao Zhu, Aochuan Chen, Haiyun Jiang, Deng Cai, Victor Wai Kin Chan, and Jia Li. Glbench: A comprehensive benchmark for graph with large language models, 2024.

[34] Michael Withnall, Hongming Chen, and Igor V Tetko. Matched molecular pair analysis on large melting point datasets: a big data perspective. *ChemMedChem*, 2018.

[35] Rodrigo Casasnovas, Joaquin Ortega-Castro, Juan Frau, Josefa Donoso, and Francisco Munoz. Theoretical pka calculations with continuum model solvents, alternative protocols to thermodynamic cycles. *International Journal of Quantum Chemistry*, 2014.

[36] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019.

[37] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, pages 28877–28888, 2021.

[38] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, page 2069–2080, 2021.

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

[40] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *ICML*, 2021.

[41] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 594–604, 2022.

[42] Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. Graphmae2: A decoding-enhanced masked self-supervised graph learner. In *Proceedings of the ACM Web Conference 2023*, page 737–746, 2023.

[43] Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng Yang, and Chuan Shi. Graphtranslator: Aligning graph model to large language model for open-ended tasks. In *Proceedings of the ACM Web Conference 2023*, 2024.

A Dataset descriptions

Table 3: Dataset statistics

Domain	Dataset	Avg. #Nodes	Avg. #Edges	#Classes/Tasks	#G
Citation	Arxiv	169,343	1,166,243	40	1
	Pubmed	19,717	44,338	3	1
	Cora	25,120	91,140	70	1
E-commerce	Computer	87,229	721,081	10	1
	Photo	48,362	500,928	12	1
	Children	76,875	1,554,578	24	1
	Sports	173,055	1,773,500	13	1
Web link	WikiCS	11,701	216,123	10	1
Knowledge graph	FB15K237	14,541	310,116	237	1
Social network	Reddit	33,434	198,438	2	1
	Instagram	11,339	144,010	2	1
Molecular	ChEMBL	25.87	55.92	1048	365,065
	HIV	25.51	54.95	1	41,127
	BACE	34.09	73.72	1	1,513
	PCBA	25.97	56.20	128	437,929
	Esol	13.29	27.35	1	1,128
	Lipo	27.04	59.00	1	4,200
	Freesolv	8.72	16.76	1	642

Arxiv Arxiv [25] is a large-scale citation graph derived from arXiv Computer Science papers. Each node corresponds to a paper and edges represent citation links between papers. The task is to classify each paper into one of 40 arXiv subcategories, such as "cs.LG" or "cs.AI". This dataset serves as a representative benchmark for large-scale node classification.

Pubmed Pubmed [29] is a citation network of biomedical research papers from the PubMed database. Each node is a paper and edges correspond to citation links. The classification task involves assigning each paper to one of three disease-related categories.

Cora The Cora [30] dataset is a citation graph where each node corresponds to a research paper, and each edge represents a citation link between papers. The dataset focuses on papers within the machine learning domain and includes 70 fine-grained categories, making the classification task particularly difficult.

Computer The Computer [26] dataset is co-purchased or co-viewed product graph, where each node represents a product in the computer category, and edges indicate that two products were frequently co-purchased or co-viewed by users. The textual content associated with each node consists of user-generated reviews for the corresponding product.

Photo The Photo [26] dataset is an e-commerce product graph where nodes represent photographic products, and edges indicate that two items were either co-purchased or co-viewed by users. The textual content of each node consists of user reviews associated with the corresponding product.

Children The Children [26] dataset is a co-purchased or co-viewed product graph focused on children's books. Nodes correspond to individual books, and edges connect books that were frequently browsed or bought together. Each node is associated with textual information including the book's title and descriptive metadata.

Sports The Sports [26] dataset is a co-purchased or co-viewed product graph in the sports domain. Nodes represent sports-related products, and edges indicate that two items were often purchased or viewed together. The associated text for each node consists of the product's title.

WikiCS WikiCS [32] is a web link network constructed from English Wikipedia articles related to computer science. Nodes are individual articles, and directed edges represent hyperlinks between them. The node text is the full content of each article.

Table 4: Task-specific descriptions used as T^{τ}_{desc}.

Task Type	**Task Description (T^{τ}_{desc})**
Node Classification	Determine this node's most likely category within the network's classification schema.
Link Prediction	Determine whether there is a specific relationship between these two nodes.
Graph Classification	Determine whether the molecule possesses specific physicochemical or bioactivity properties.
Graph Regression	Predict the continuous numerical value of a physicochemical or bioactivity property of the molecule.

FB15K237 FB15K237[13] is a large-scale knowledge graph where each node represents an entity (e.g., a person, location, or object) and each edge corresponds to a relational triple connecting two entities. Textual content for nodes is constructed from entity names and relation descriptions.

Reddit The Reddit[33] dataset is a social interaction graph where nodes correspond to users and edges represent reply interactions in threads from a specific time period. Node-associated text consists of user posts from Reddit threads.

Instagram Instagram [33] is a social graph in which each node represents a user, and edges denote social connections such as following relationships. The textual content associated with each node is extracted from users' self-introductions or profile descriptions.

ChEMBL ChEMBL [27] is a molecular graph dataset where each graph corresponds to a chemical compound. Nodes represent atoms, and edges denote chemical bonds. The textual information for each molecule is given by its SMILES (Simplified Molecular Input Line Entry System) representation.

HIV The HIV [31] dataset consists of molecular graphs representing candidate compounds for HIV treatment. Nodes denote atoms and edges are chemical bonds. Each molecule is described by its SMILES string.

BACE BACE [31] is a molecular dataset used in bioactivity classification. Each graph is a molecule, with atoms as nodes and chemical bonds as edges. SMILES strings provide the molecular structure information in text format.

PCBA PCBA [31] is a large-scale molecular dataset for virtual screening. Each graph is a molecule, modeled by atoms and bonds, with SMILES strings representing the underlying chemical structure.

Esol The Esol [34] dataset contains water-solubility data for chemical compounds. Each molecule is modeled as a graph, with node and edge structures corresponding to atoms and bonds. SMILES strings serve as the textual representation.

Lipo Lipo is a molecular dataset focused on lipophilicity prediction. Each molecule is represented as a graph with atoms as nodes and bonds as edges. The SMILES string encodes each molecule's structure in text form.

Freesolv Freesolv [35] consists of molecular graphs used for estimating hydration free energy. Each molecule is modeled by a graph of atoms and bonds. The SMILES representation is used as the text-based molecular description.

B Details of prompt descriptions

For each specific task τ, we define a high-level textual description T^{τ}_{desc} that serves as the general instruction for the task. This description is task-type dependent and is designed to guide the model's

Table 5: Task-specific instruction templates used as T^{τ}_{detail}.

Dataset	Instruction Template (T^{τ}_{detail})
Arxiv, Pubmed, Cora	Given a representation of a paper with the following information: Title: {title}, Abstract: {abstract}. Question: Which arXiv CS sub-category does this paper belong to? Please directly give the most likely answer from the following sub-categories: {candidate_labels}.
Children	Given a representation of a book with the following information: Name: {title}, Content: {abstract}. Question: Which category does this book belong to? Please directly give the most likely answer from the following categories: {candidate_labels}.
Computer, Photo	Given a representation of a book with the following information: Name: {title}, Content: {abstract}. Question: Which category does this book belong to? Please directly give the most likely answer from the following categories: {candidate_labels}.
Sports	Given a representation of an electronic product with the following information: Comment: {comment}. Question: Which category does this electronic product belong to? Please directly give the most likely answer from the following categories: {candidate_labels}.
WikiCS	Given a representation of a Wikipedia page with the following information: Name: {name}, Content: {content}. Question: Which category does this Wikipedia page belong to? Please directly give the most likely answer from the following categories: {candidate_labels}.
Reddit	Given a representation of a user with the following information: Previous posts: {posts}. Question: Which category does this user belong to? Please directly give the most likely answer from the following categories: {candidate_labels}.
Instagram	Given a representation of a user with the following information: Personal introduction: {introduction}. Question: Which category does this user belong to? Please directly give the most likely answer from the following categories: {candidate_labels}.
FB15K237	Given the representation of two entities: First entity: Name: {name}, Description: {description}. Second entity: Name: {name}, Description: {description}. Question: Which category should the relation between these two entities be classified as? Please directly give the most likely answer from the following categories: {candidate_labels}.
Arxiv-link, Pubmed-link	Given the representation of two papers: Title: First Paper: {title}, Second Paper: {title}. Question: Do these two papers have citation relationships? Please choose the most likely answer from: "Yes, they have citation relationships" or "No, they do not have citation relationships".
Children-link	Given the representation of two books: Title: First Book: {title}, Second Book: {title}. Question: Do these two books have co-purchased or co-viewed relationships? Choose from: "Yes, they have co-purchased or co-viewed relationships" or "No, they do not have co-purchased or co-viewed relationships".
Computer-link, Photo-link	Given the representation of two electronic products: Title: First Product: {comment}, Second Product: {comment}. Question: Do these two products have co-purchased or co-viewed relationships? Choose from: "Yes, they have co-purchased or co-viewed relationships" or "No, they do not have co-purchased or co-viewed relationships".
ChEMBL, HIV, BACE, PCBA	Given a representation of a molecule with the following information: SMILES: {smiles}. Question: {task} Please answer: "Yes, this molecule is effective to this assay" or "No, this molecule is not effective to this assay".
Esol, Lipo, Freesolv	Given a representation of a molecule with the following information: SMILES: {smiles}. Question: {task} Please provide a single numerical value rounded to two decimal places.

understanding and response generation. The full list of task descriptions used in our experiments is shown in Table 4.

Additionally, we provide task-specific content instructions T^{τ}_{detail}, which include a description of the graph content and the corresponding question. The details are shown in the table 5.

C Details of experimental setup

Datasets For data splitting, we follow standard splits for node classification, graph classification, and graph regression tasks. For link prediction, we randomly split the data into training/validation/test sets with a ratio of 8:1:1. To ensure fair comparison, all baseline models are evaluated using the same splits. Due to the large scale of the training data, we sample a subset of instances from each training dataset. Specifically, we sample 45,470 instances from **Arxiv**, 21,888 from **Children**, 31,378 from **Computer**, 10,000 each from **Arxiv-link**, **Children-link**, and **Computer-link**, 29,440 from **FB15K237**, and 74,242 from **ChEMBL**.

Table 6: In-domain zero-shot results.

Model	Pubmed	Cora	Photo	Sports	BACE	HIV	PCBA	Pubmed	Photo
	Node Classification				Graph Classification			Link Prediction	
w/o AT	0.721	0.155	0.384	0.371	0.492	0.467	0.497	0.502	0.576
w/o TA	0.766	0.211	0.399	0.311	0.487	0.499	0.489	0.609	0.710
UniGTE	**0.870**	**0.215**	**0.565**	**0.403**	**0.534**	**0.501**	**0.541**	**0.722**	**0.732**

Table 7: Zero-shot performance on cross-domain node classification and cross-task graph regression.

Setting	WikiCS	Reddit	Instagram	Esol	Lipo	Freesolv
	Node Classification (Accuracy)			Graph Regression (MAE)		
w/o AT	0.290	0.309	0.391	6.58	11.22	64.11
w/o TA	0.645	0.502	0.590	8.35	2.44	12.37
UniGTE	**0.680**	**0.510**	**0.601**	**2.54**	**1.03**	**9.18**

Table 8: Legality rate (%) across models and datasets

Dataset	Pubmed	Cora	Photo	Sports	WikiCS	Reddit	Instagram	HIV	BACE	PCBA	Esol	Lipo	Freesolv
Model	Legality rate (%)												
Vicuna-7B	**100**	**95.8**	94.1	99.6	64.7	57.4	90.3	96.0	92.1	39.0	99.1	81.2	98.5
LLaGA	**100**	76.5	83.8	99.2	99.0	99.0	99.5	82.1	**100**	75.1	66.3	82.2	57.0
TEA-GLM	**100**	95.6	96.2	**100**	70.9	96.8	99.7	**100**	**100**	**100**	80.6	88.1	45.0
UniGTE	**100**	95.0	**99.7**	99.7	**99.6**	**100**	**100**	**100**	**100**	**100**	**99.1**	**99.8**	**100**

Baselines For **LLaGA** and **TEA-GLM**, we re-run their training pipelines under our experimental settings using Vicuna-7B as the predictor. As **GOFA** requires extensive pretraining, we directly use the officially released pretrained checkpoint and conduct instruction tuning under our experimental setup. For all baselines, we follow the hyperparameter configurations provided in their respective original papers.

Training details We use **BERT** as the pretrained language model (PLM), and **Vicuna-7B** as the large language model (LLM). In our implementation of UniGTE, we set the LoRA learning rate and the MLP learning rate to 2e-4, while the learning rate for the graph relative position embedding is 2e-3. We use a batch size of 2, apply gradient clipping with a maximum norm of 10, and perform gradient accumulation every 2 steps. The number of alignment tokens is fixed to 64 across all experiments. All experiments are conducted on a machine with two NVIDIA A100 GPUs, each equipped with 80GB of memory.

D More experimental results

D.1 Ablation study details

We provide detailed ablation results for each dataset. The in-domain results are presented in Table 6, while the cross-domain and cross-task results are shown in Table 7. As the tables indicate, removing the alignment tokens significantly degrades the model's transferability. Additionally, replacing the task-specific description with a fixed, generic prompt leads to performance drops across different datasets. These findings validate the effectiveness of the proposed alignment tokens in enhancing generalization, and further demonstrate the importance of incorporating task-specific signals into the encoding process.

D.2 Legality rate

The training process may affect the instruction-following ability of LLMs in zero-shot scenarios. Specifically, while the model can generate appropriate outputs on the training set, it often fails to produce legality answers on unseen datasets. This essentially reflects poor generalization ability.

Following the approach proposed in [43], we use the *legality rate* to measure the proportion of valid responses generated by the model on unseen datasets.

As shown in Table 8, existing baselines exhibit poor instruction-following performance on certain datasets. In contrast, our model consistently maintains strong instruction compliance across all unseen datasets, generating valid answers. This further demonstrates the superior generalization capability of our approach.

D.3 Efficiency and computational complexity analysis

We have conducted a comparison between LLM and UniGTE under a zero-shot setting using equivalent graph information. The results are summarized in the tables below, where the first table reports inference time and the second table presents task-specific performance metrics, showing that UniGTE achieves significantly faster inference speed while maintaining superior accuracy.

Table 9: Inference time compared with the LLM

Sec / sample (s)	Pubmed	Cora	PCBA	Photo_link	WikiCS	Instagram	Lipo
Vicuna-7B	0.8	2.6	3.2	1.7	0.7	3.3	10.2
UniGTE	0.5	0.6	1.0	1.1	0.3	0.3	2.0
Improvement	37.5%	78.7%	70.3%	34.1%	50.8%	91.7%	80.5%

Table 10: Task performance compared with the LLM

Task Metric	Pubmed	Cora	PCBA	Photo_link	WikiCS	Instagram	Lipo
Vicuna-7B	0.721	0.155	0.497	0.576	0.290	0.391	11.220
UniGTE	0.870	0.215	0.541	0.732	0.680	0.601	1.030
Improvement	20.7%	38.7%	8.9%	27.1%	134.5%	53.7%	90.8%

Here, we compare the time complexity of GOFA. We adopt the same notation used in the GOFA paper: suppose a graph contains V nodes, E edges, and each node is represented by k tokens on average. In GOFA, each node is expanded into k tokens ($k \gg 1$), and the self-attention is restricted within each node's token group. Thus, the per-layer time complexity is: $\mathcal{O}(k \cdot k \cdot V) = \mathcal{O}(Vk^2)$, which is consistent with the complexity reported in the original GOFA paper. In contrast, UniGTE represents each node using a single token and performs cross-node attention directly. Therefore, the complexity becomes: $\mathcal{O}((V \cdot 1)^2) = \mathcal{O}(V^2)$. This is considerably lower, especially when $k = 128$ or higher as required in GOFA for satisfactory performance. Furthermore, our method leverages k-hop subgraph sampling, which keeps the number of nodes V relatively small, making $\mathcal{O}(V) \ll \mathcal{O}(k^2)$ and thus $\mathcal{O}(V^2) \ll \mathcal{O}(Vk^2)$.

In addition to the Transformer layers of the LLM itself, GOFA also introduces external GNN layers. This component is not accounted for in its original complexity analysis. Assuming the average node degree is d, the GNN layer introduces an additional cost of $\mathcal{O}(V \cdot d \cdot k)$. Combining all these considerations, our method provides a clear computational advantage over GOFA.

To further validate this, we compare per-sample inference time and accuracy under the zero-shot setting between GOFA and UniGTE. As shown in the table below, UniGTE achieves faster inference and better performance, demonstrating the efficiency and effectiveness of our design.

Table 11: Inference time compared with GOFA

Sec / sample (s)	Pubmed	Cora	PCBA	Photo_link	WikiCS	Instagram	Lipo
GOFA	1.4	5.1	1.5	1.8	2.0	1.2	2.1
UniGTE	0.5	0.6	1.0	1.1	0.3	0.3	2.0
Improvement	63.2%	89.3%	34.5%	38.5%	83.6%	77.1%	5.7%

Table 12: Task performance compared with GOFA

Task Metrics	Pubmed	Cora	PCBA	Photo_link	WikiCS	Instagram	Lipo
GOFA	0.614	0.039	0.500	0.504	0.613	0.367	1.360
UniGTE	0.870	0.215	0.541	0.732	0.680	0.601	1.030
Improvement	41.7%	451.3%	8.2%	45.2%	10.9%	63.8%	24.3%

Table 13: Supervised results under SFT setting

Model	Arxiv	Children	Computer	Arxiv_link	Children_link	Computer_link	FB15K237
GraphGPT-std	0.625	-	-	-	-	-	-
LLaGA	0.711	**0.498**	0.579	0.901	**0.837**	**0.793**	0.870
TEA-GLM	0.661	0.477	0.549	0.891	0.767	0.738	**0.910**
GOFA	0.659	0.190	0.514	0.772	0.505	0.563	0.731
UniGTE	**0.713**	0.431	**0.589**	**0.913**	0.813	0.765	0.901

D.4 Supervised results

We have conducted additional experiments under the SFT setting and compared UniGTE with relevant baselines that use LLMs as predictors. The results are presented below.

As shown in Table 13, UniGTE achieves competitive performance and yields strong results on most datasets, even surpassing state-of-the-art baselines on several of them. This highlights its promising capability under SFT settings. These results demonstrate that although our model is originally designed for zero-shot scenarios, it also performs robustly under full supervision, without significant degradation compared to existing approaches.

D.5 Graph understanding tasks

We have conducted additional experiments on graph understanding tasks to demonstrate our model's graph understanding/reasoning capabilities, and compared the results with relevant baselines.

Table 14: Comparison of various models on graph reasoning tasks

Model	Pubmed_CONN	Cora_CONN	Pubmed_SPD	Cora_SPD	Pubmed_CN	Cora_CN	BACE_CYCLE	HIV_CYCLE	PCBA_CYCLE
Vicuna-7B	0.551	0.505	1.51	2.53	5.07	12.85	2.04	1.37	2.11
LLaGA	0.618	0.553	3.72	7.98	-	-	-	-	-
TEA-GLM	0.689	0.623	3.06	2.34	5.27	7.79	6.94	7.39	9.74
GOFA	0.638	**0.694**	1.43	1.68	**1.13**	7.24	6.96	5.58	6.24
UniGTE	**0.707**	0.616	**1.11**	**1.42**	1.20	**2.24**	**1.83**	**1.10**	**1.92**

We directly evaluated the model trained in our original submission on zero-shot graph understanding tasks, without any fine-tuning on these tasks. All other baselines were evaluated under the same zero-shot setting, except for GOFA, which incorporates some graph understanding tasks during its pretraining phase. To ensure fair comparison, all methods were tested on the same test sets. Following the experimental setup of GOFA and the references you kindly suggested, we constructed the following tasks and generated the data using real-world datasets. We used AUC for evaluating the CONN task, and MAE for the other tasks.

- CONN: Determine whether two nodes are connected
- SPD: Predict the shortest path distance between two nodes
- CN: Predict the number of common neighbors between two nodes
- CYCLE: Predict the number of cycles in the graph

The results are summarized in the Table 14, where the best result is bolded and the second-best is italicized. Our method achieves state-of-the-art performance on most datasets. Notably, compared to Vicuna, which serves as our base model, our method consistently outperforms it across all tasks and datasets—highlighting the strong generalization and zero-shot capabilities of our approach.

Compared to other baselines, our method is either the best or competitive. It is worth noting that GOFA, benefiting from the inclusion of graph understanding tasks during its pretraining phase, performs relatively well on a few specific datasets. In contrast, our method has never encountered any graph understanding tasks during training, making the evaluation a significantly more challenging and rigorous zero-shot setting. Despite this, our approach still delivers consistently strong performance, further demonstrating its generalization capability.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We verify the contributions of our proposed method through experiments, and the results in the Sec. 3 effectively demonstrate the contributions we outlined in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the current shortcomings of our method and future research directions in Sec.5

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide all necessary information to ensure the reproducibility of our main experimental results. Section 2 details the proposed methodology, while Sections 2.3 and detailed setting in Appendix C outline the training procedures and implementation settings. All reported results in this paper can be reliably reproduced based on the disclosed configurations.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We use publicly available datasets in all experiments. To ensure reproducibility, we will include the complete source code in the supplementary materials in an anonymized form.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide details on data splits, hyperparameter settings, and baseline configurations in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: Due to the high computational cost associated with training large language models (LLMs), we do not report error bars or variance across multiple runs. However, our evaluation spans a wide range of datasets and task types across multiple domains, and the consistent performance of our method across these diverse settings provides strong empirical support for the robustness and generalization of the proposed approach.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the computer resources in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that our work complies with the NeurIPS Code of Ethics, including ethical standards for data collection, usage, and experimentation.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The paper does not discuss societal impacts because it focuses on foundational research in graph learning. The method is generic and does not directly involve application scenarios with immediate societal implications.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release any data or models that pose high risk, and this work does not involve or introduce such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly cite the original papers corresponding to all models and datasets used in our work. Wherever applicable, we also ensure that we comply with the licenses and terms of use of these assets. For publicly available datasets and pretrained models, we use only those released under permissible licenses (e.g., MIT, CC-BY).

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Our paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: This paper employs a large language model (Vicuna-7B) as a core component of the proposed encoder-decoder framework. Specifically, the LLM is used as the decoder to generate task-specific outputs conditioned on alignment tokens. The LLM's generalization ability is integral to the design of our method and plays a central role in achieving zero-shot performance across diverse graph tasks.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.