CELLM: Curvature Enhanced Large Language Models for Graph Structure Learning

Xin Ding, Jiaxin Ding ✉, Haochen Han, Haoxiang Zhang, Yule Xie, Luoyi Fu, and Xinbing Wang

Shanghai Jiao Tong University
{DiligentDing, jiaxinding, haochen_neyc, 1227538797, xyl-alter, yiluofu, xwang8}@sjtu.edu.cn

Abstract. In recent years, Large Language Models (LLMs) have achieved remarkable success across various domains, sparking interest in extending their capabilities to graph structured data. However, leveraging LLMs for graph data poses significant challenges due to the inherent mismatch between graph and text modalities. Existing approaches primarily rely on two strategies: Graph-to-Text Translation, which describes graph structures in natural language to enable LLMs to process structural information, and Graph-to-Token Conversion, which transforms graphs into sequences of tokens aligned with text tokens. Although these methods have achieved a certain degree of integration between LLMs and graph data, they still struggle to fully capture the complex structure in real-world graphs and fail to provide a global view of the long-range dependencies—the relationships or interactions between nodes that are far apart in the graph. To bridge this gap and enhance LLM's understanding of the graph structure, we propose Curvature Enhanced Large Language Model (CELLM), a novel architecture that integrates advanced graph structural information with discrete graph curvature to offer a global and geometric perspective. Graph curvature leverages local curvature measurements to derive global insights, such as assessing overall connectivity, identifying bottlenecks, or detecting hubs. This more expressive structural measure can provide LLMs with enhanced capabilities to capture and comprehend complex graph structures. Additionally, we implement a task-specific tuning procedure to further improve the structure understanding within LLMs. Extensive experiments demonstrate the effectiveness of our proposed CELLM across graph-related tasks, highlighting its potential in improving the expressiveness and understanding of LLMs when applied to graph modalities.

Keywords: Large Language Model, Graph Curvature, Graph Structure Learning

1 Introduction

Large Language Models (LLMs), such as GPT [15], have demonstrated remarkable success in various natural language processing (NLP) tasks. Motivated by these achievements, there is a growing interest in the application of LLMs to graph structured data [6, 5, 17]. As graphs are fundamental data structures used to model complex relationships and interactions in various domains, integrating LLMs with graph data holds the potential to harness their powerful reasoning and understanding abilities for graph related tasks.

However, applying LLMs to graph data poses significant challenges due to the mismatch between the text and graph modalities [9]. Graphs are non-Euclidean and exhibit complex topologies with rich structural information (e.g., hubs and bottlenecks), and long-range dependencies where distant nodes can influence each other through indirect connections. These characteristics are not naturally accommodated by language models designed for sequential text data, and limits the ability of LLMs to fully learn the intricate relationships present in graph data. Existing works address this issue through two main strategies: (1) **Graph-to-Text Translation**: This approach translates graph structure into nature language descriptions [20, 2]. By encoding graph properties, relations, and patterns into text formats through refined prompting and tuning techniques, LLMs can learn the structure information with language understanding capabilities. However, for large-scale graphs, the translated text description tends to exceed the context window of LLMs, leading to loss of information and inaccurate description. Moreover, such descriptions are typically verbose and focus on local edge connection details [3], providing limited insight into the global structure, which constrains LLM's capability to capture interactions between nodes that are far apart [9]. (2) **Graph-to-Token Conversion**: This approach converts graphs into token sequences and aligns them with text tokens. Recent works employ Graph Neural Networks (GNNs) to generate graph structure-aware embeddings as a prefix [6, 12], that are fed into LLMs for downstream tasks for alignment. While these methods help incorporate graph structure into LLMs to some extent, they still face challenges in enabling LLMs to fully grasp the complex relationships within large-scale graphs. The primary issue lies in the limited expressive power of GNN-based representations, which struggle to capture the long-range, global structure of graphs, due to issues such as over-smoothing and limited receptive fields [21].

In summary, existing methods fail to enable LLMs to capture and comprehend complex graph structure information. The fundamental challenge lies in bridging the gap between the non-Euclidean, multi-dimensional nature of graph data and the sequential, one-dimensional processing paradigm of LLMs. Therefore, there is a pressing need for a more expressive architecture that incorporates advanced graph structural information from a geometric perspective to address these limitations.

To address this challenge, we propose **C**urvature **E**nhanced **L**arge **L**anguage **M**odels (CELLM) , a novel architecture that leverages advanced geometric discrete graph Ricci curvature to enhance LLM's ability to capture and understand the complex structure information within graphs. Graph curvature provides a way to quantify "shape" of the graph, capturing both local and global properties, such as connectivity robustness, clustering tendencies, and the presence of bottlenecks and hubs[21, 18]. Thus, we can employ curvatures to enhance LLM's capability to better capture and represent complex structures in graphs. Specifically, we exploit the curvature information to reweigh different channels of the messages to obtain more expressive and informative graph embeddings and align them with text space. Additionally, we incorporate task-specific tuning procedures including prompt tuning and Low-Rank Adaptation (LoRA) techniques to further improve LLM's ability to understand and process graph structures effectively. Extensive experiments validate the effectiveness of the proposed CELLM, demonstrating its capability to enhance LLM's comprehension of structural information within graph data.

Our main contributions can be summarized as follows:

- To the best of our knowledge, we are the first to integrate LLMs with graphs from advanced geometric curvature perspective.
- We propose a new framework, CELLM, which exploits discrete curvature to enhance LLM's ability to capture and comprehend the intricate structures contained in the graph.
- Extensive experiments across various graph benchmarks validate the effectiveness of our framework, showcasing its enhanced performance on node classification tasks.

2 Methodology

We propose CELLM, a novel framework designed to enhance the LLM's capability to capture and comprehend the graph structure through two core components: Curvature Graph Projector and Task-Specific LLM Prompt Tuning. The overall architecture of CELLM is depicted in Figure 1.

2.1 Curvature Graph Projector

In large-scale real-world graphs, a variety of geometric structures, including grids, circles, trees and cliques, frequently appear simultaneously within a unified framework. It is intractable to accurately capture these complex structures through the node degree as different graph topologies can share identical degree distributions. And the traditional message passing schemes can only provide local structure. To overcome these limitations and enhance LLM's capability to grasp the intricate topologies contained in graphs, we propose Curvature Graph Projector, inspired by work [21], which consists of two components: Ricci curvature computation, Curvature Graph Convolution and Alignment.

Ricci curvature computation To generalize Ricci curvature to discrete spaces, Ollivier [13] proposed a coarse approximation using a probability measure m_x of total mass 1 around x. Thus the distance can be measured by Wasserstein distance, which finds the optimal transportation plan that preserves mass between two probability measures. Then the Ricci curvature κ_{xy} on edge (x, y) can

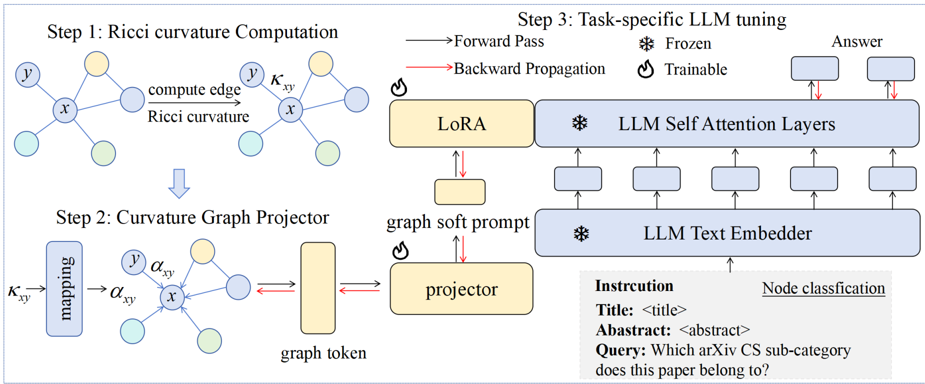

Fig. 1: Framework of our proposed CELLM. Our method enhances LLM's capability to comprehend graph structure through a three-step process: (1) Ricci curvature Computation (2) Curvature Graph Projector (3) Task-specific LLM Prompt Tuning

be defined as $\kappa_{xy} = 1 - \frac{W(m_x, m_y)}{d(x,y)}$, where $W(m_x, m_y)$ denotes the Wasserstein distance and $d(x,y)$ denotes the length of the shortest path between node x and node y.

In the metric space, the concept of a small ball S_x at point x corresponds to the 1-hop neighborhood $N(x)$ of node x in a graph. This analogy serves as the foundation for defining the Ollivier-Ricci curvature on graph edges. For a graph $G = (V, E)$, a probability measure m_x^α at node x can be defined as:

$$m_x^\alpha(x_i) = \begin{cases} \alpha & \text{if } x_i = x, \\ \frac{1-\alpha}{|N(x)|} & \text{if } x_i \in N(x), \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where $N(x)$ represents the 1-hop neighborhood of node x and $|N(x)|$ denotes the number of neighbors and α is a parameter within [0,1] that represents the probability mass retained at node x itself, and the rest is distributed uniformly across its neighbors. We set the parameter $\alpha = 0.5$, as suggested in [14]. As for the computation of the Wasserstein distance $W(m_x^\alpha, m_y^\alpha)$ between the probability measures around two endpoints x and y of the edge (x, y), the optimal transportation plan can be solved by the linear programming:

$$\min_M \sum_{i,j} d(x_i, y_j) M(x_i, y_j) \tag{2}$$

subject to:

$$\sum_j M(x_i, y_j) = m_x^\alpha(x_i), \forall i; \ \sum_i M(x_i, y_j) = m_y^\alpha(y_i), \forall j \tag{3}$$

where $M(x_i, y_j)$ represents the amount of probability mass transported from node x_i to y_j along the shortest path with distance $d(x_i, y_j)$. The definition of Ricci curvature highlights its capability to capture graph structure from a global and geometric perspective, effectively describing long-range dependencies within the graph.

Curvature Graph Convolution and Alignment In this section, we introduce how Ricci curvature can be leveraged to capture complex structures and detail the alignment procedure. Intuitively, curvature reflects how easily message propagates through edges [18]. If the curvature is positive, it indicates that more information can propagate through the edge, vice versa. Thus we consider exploiting the Ricci curvature to guide the message passing procedure in the graph convolution. Specifically, we adopt a data-driven approach by learning a mapping function that translates curvature values into weights for message propagation. To be specific, the graph convolution process can be formulated as follows:

$$H_x^{t+1} = \sigma(\sum_{y \in N(x)} \alpha_{xy}^t W^t H_y^t) \tag{4}$$

where H_x^t denotes the hidden state embedding of node x at step t and α_{xy} represents the attention coefficient to reweigh the message. To enhance the flexibility and expressiveness of the model, we map Ricci curvature values to attention coefficients using a multi-valued approach. Specifically, we employ a multi-layer perceptron (MLP) to approximate the mapping function, converting the Ricci curvature κ_{xy} into a reweighting vector $\boldsymbol{\alpha}_{xy}^t \in \mathbb{R}^{D^{t+1}}$. This approach allows us to assign different weights to individual message channels. Subsequently, we apply a channel-wise softmax function to normalize the mapping outputs separately on each message channel, formally $\boldsymbol{\alpha}_{xy} = \mathrm{softmax}(\mathrm{MLP}(\kappa_{xy}))$. And for the curvature graph convolution, the message passing and updating process can be formulated as follows:

$$H_x^{t+1} = \sigma(\sum_{y \in N(x)} \mathrm{softmax}(\mathrm{MLP}^t(\kappa_{xy}))W^t H_y^t) \tag{5}$$

This formulation leverages curvature-informed reweighting to enhance the model's capacity to capture complex graph structures and provide long-range dependencies from a global and geometric view.

Furthermore, it is critical to align the graph token generated through curvature with the textual description token. To achieve this, we employ a lightweight multi-layer perceptron as the projector module to map the learned graph tokens into the space of text tokens. The aligned graph token can be viewed as graph-based soft prompt and serve as trainable prefixes to the textual input, encapsulating enriched graph structural information.

2.2 Task-specific LLM Prompt Tuning

After leveraging Ricci curvature to capture the complex structures contained in graph from a global and geometric view, the CELLM proceeds to process the textual information and employ prompt tuning techniques to incorporate the learned graph structure information.

The generation process, incorporating the prompt tuning procedure, can be formulated as follows:

$$P_{\theta,\phi}(Y|G,Q) = \prod_{i=1}^{m} p(y_i|y_{<i}, [E_G; E_D; E_Q]) \tag{6}$$

where θ denotes the fixed parameters of the pretrained LLM, ϕ represents the learnable parametres of the prompt tuning projector, E_G denotes the aligned graph structure embedding derived from Ricci curvature, E_D represents the textual embedding of the language description of the nodes, and E_Q represents the textual embedding of the query designed for the specific downstream tasks. These embeddings are concatenated and fed into the LLM to generate the response $Y = \{y_1, y_2, \ldots, y_m\}$, where m is the sequence length. The entire framework is trained in an end-to-end manner, optimizing the learnable parameters ϕ of the prompt tuning module to align the LLM's outputs with the ground truth.

To further enhance the LLM's capability to comprehend the graph structural information, we integrate Low-Rank Adaptation (LoRA) [7] with prompt tuning.By leveraging low-rank matrix decomposition, LoRA significantly reduces the number of trainable parameters required.By integrating prompt tuning with LoRA, our framework enables the model to efficiently incorporate graph structural knowledge while maintaining the adaptability needed for diverse downstream tasks. This approach provides a lightweight yet effective approach to leveraging both graph and textual data.

3 Experiments

3.1 Experiment Setups

Datasets We train and evaluate our proposed model on five widely recognized graph datasets: Cora, Citeseer, Pubmed [16], Arxiv-2023 [5] and ogbn-Products [8]. These datasets encompass domains such as citation networks and e-commerce, varying in terms of sparsity and size, ranging from small to large scales. In citation graphs (Cora, Citeseer, Pubmed, Arxiv-2023), nodes correspond to research papers, with textual features derived from the titles and abstracts, while edges represent citation relationships among the papers. For ogbn-Products, nodes represent Amazon products characterized by item descriptions, and edges capture co-purchase relationships. Our experiments mainly focus on node classification tasks.

Baselines We evaluate our proposed method against a range of baseline approaches, encompassing both traditional graph learning techniques and methods leveraging Large Language Models (LLMs):

– **GNN-based methods**: This category includes well-established Graph Neural Network (GNN) architectures, including Graph Convolutional Networks (GCN)[10], Graph Attention Networks (GAT) [19], GraphSAGE [4] and RevGAT [11].
– **LLM-only methods**: We consider approaches that process graph information directly as textual descriptions using LLMs. This category includes implementations utilizing zero-shot inference, prompt tuning , and Low-Rank Adaptation (LoRA) techniques.
– **Hybrid GNN-LLM methods**: We compare our approach with state-of-the-art methods that integrate GNNs and LLMs. Specifically, we include GraphGPT [17] and GraphPrompter [12], which focus on text-attributed graph representation learning with language models. Additionally, we use LLaGA [1] as a baseline, which transforms graph structure into node sequences with two type of templates: Neighborhood Detail Template (ND) and Hop-Field Overview Template (HO).

Table 1: Node classification results of our proposed CELLM, with a number of baselines under various graph benchmarks: Cora, Citeseer, Pubmed, Arxiv-2023 and ogbn-Products. The highest-performing results are highlighted in bold, while the second-best results are marked with an underline.

Method	Cora	Citeseer	Pubmed	Arxiv-2023	ogbn-Products
GCN	0.8147	0.7172	0.8521	0.7295	0.7132
GAT	0.8099	0.7078	0.8409	0.7182	0.7052
GraphSAGE	0.8263	0.7170	0.8413	0.7019	0.7131
RevGAT	0.8353	0.7274	0.8502	0.7083	0.7189
zero-shot	0.4331	0.2922	0.9139	0.4423	0.1505
prompt tuning	0.7031	0.7097	0.9145	0.7199	0.7514
LoRA	0.7597	0.7345	0.9413	0.7458	0.7899
GraphGPT	0.7862	0.6782	0.7416	0.6390	0.6713
GraphPrompter	0.8026	0.7361	<u>0.9480</u>	<u>0.7561</u>	<u>0.7954</u>
LLaGA-ND	0.8432	0.7572	0.9406	0.7535	0.6519
LLaGA-HO	<u>0.8545</u>	<u>0.7587</u>	0.9459	0.7479	0.7421
CELLM(ours)	**0.8653**	**0.7648**	**0.9516**	**0.7867**	**0.8198**

3.2 Main results

Table 1 presents the results of node classification experiments. Across various graph benchmarks, our proposed model, CELLM, consistently outperforms other baseline methods. These results underscore the effectiveness and superiority of CELLM in addressing graph learning tasks. And we can observe that zero-shot approach consistently achieves the lowest performance across all datasets, highlighting the limitations of large language models (LLMs) in capturing and understanding the intricate structures inherent to graphs. Moreover, CELLM outperforms all pure GNN and LLM baselines across all datasets, which demonstrates the necessity of integrating textual and graph structural information when dealing with text attributed graph tasks. Furthermore, our proposed model generally surpasses the performance of hybrid GNN-LLM methods, particularly on large-scale graphs such as Arxiv-2023 and Ogbn-products. These large-scale graphs, with more complex structures, pose greater challenges. The superior performance of CELLM on these large-scale graphs indicates its enhanced ability to handle complex real-world graph structures effectively.

In conclusion, our proposed CELLM demonstrates the improved performance across various datasets on node classification tasks across diverse datasets. The improvements over the the existing hybrid GNN-LLM baselines prove its capability to empower LLMs to better capture and understand the complex graph structures, especially in large-scale graph scenarios.

3.3 Ablation Studies

In this section, we conduct comprehensive ablation studies to assess the contribution of each component in CELLM.

CELLM incorporates specific designed components, including the graph structure embedding derived from Ricci curvature and strategies for LLMs to integrating structural information into LLMs: prompt tuning and low-rank adaptation (LoRA). Additionally, we utilize a multi-layer perception to transform Ricci curvature into the attention weights. To evaluate the impact of these components, we test various model variants on node classification tasks across five datasets as follows:

- **w/o Cur**: We exploit GNNs such as GAT to obtain graph structural information without using Ricci Curvature.
- **w/o Map**: We directly exploit the curvature to weigh messages without multi-channel MLP mapping function.
- **w/o GE**: The CELLM without the graph structural embedding derived from the Ricci curvature.
- **w/o LoRA**: The CELLM without the LoRA module, keeping the LLM parameters fixed.

Table 2 shows the results of the ablation study, which evaluates the impact of the removing individual component from the proposed model. The consistent performance drops across all datasets, which underscores the significance and complementary nature of these components. In particular, we observe that the structural information captured by the Ricci curvature is critical for the CELLM. When we remove the Ricci curvature information, the performance undergoes significant decrease. This demonstrates the critical role of Ricci curvature in CELLM, as it effectively captures intricate graph structures and provides global insights with long-range dependencies. These findings emphasize the superior expressiveness of Ricci curvature in modeling complex graph information.

Table 2: Performance comparison of CELLM variants across five datasets. Best results for each dataset are in bold.

Variant	Cora	Citeseer	Pubmed	Arxiv-2023	ogbn-Products
CELLM	**0.8653**	**0.7648**	**0.9516**	**0.7867**	**0.8198**
w/o Cur	0.7864	0.7104	0.9439	0.7224	0.7850
w/o Map	0.7915	0.7254	0.9462	0.7646	0.7841
w/o GE	0.7472	0.7004	0.9470	0.7348	0.7863
w/o LoRA	0.7829	0.7073	0.9434	0.7341	0.7818

3.4 Impact of Hyperparameters

We investigate the impact of key hyperparameters in CELLM: the probability mass retained at node x itself α. Table 3 illustrates the relationship between the hyperparameter and the node classification accuracy across the graph datasets. We can observe that variations in α have a minimal effect on classification accuracy. This observation can be attributed to the design of the Ricci curvature, which is intended to capture the complex structures from a global and geometric view. As a result, changes in the probability mass α do not significantly change the inherent structural information embedded within the graph. This robustness highlights the effectiveness of the Ricci curvature in capturing the complex structures and providing long-range dependencies.

Table 3: Performance comparison of different hyperparameters across five datasets. Best results for each dataset are in bold.

Parameter	Cora	Citeseer	Pubmed	Arxiv-2023	ogbn-Products
$\alpha = 0.3$	0.8493	**0.7694**	0.9492	0.7733	0.8017
$\alpha = 0.5$	**0.8653**	0.7648	**0.9516**	0.7867	**0.8198**
$\alpha = 0.7$	0.8517	0.7405	0.9485	**0.7910**	0.8121

4 Conclusion

In this paper, we propose CELLM, a novel framework that leverages the graph curvature to enhance LLM's capability to capture and comprehend the graph structure. Instead of transforming graph structure into textual description, CELLM exploits the discrete Ricci curvature to capture the complex structures contained in the graph from a global and geometric view and enables the LLMs to fully understand the graph structural information. We conduct extensive experiments on a variety of real-world text attributed graphs and demonstrate that CELLM can outperform the existing works, which proves the effectiveness of CELLM to enable LLMs to better understand and exploit the graph structural information.

Acknowledgment

This work was supported by the National Key Research and Development Plan No. 2022YFB3904204, NSF China under Grant No. 62202299, T2421002, 62020106005, 62061146002, 62272301, Shanghai Natural Science Foundation No. 22ZR1429100.

References

1. Chen, R., Zhao, T., Jaiswal, A., et al.: Llaga: Large language and graph assistant (2024), https://arxiv.org/abs/2402.08170
2. Fatemi, B., Halcrow, J., Perozzi, B.: Talk like a graph: Encoding graphs for large language models (2023), https://arxiv.org/abs/2310.04560
3. Guo, J., Du, L., Liu, H., et al.: Gpt4graph: Can large language models understand graph structured data ? an empirical evaluation and benchmarking (2023), https://arxiv.org/abs/2305.15066
4. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NeurIPS (2017)
5. He, X., Bresson, X., Laurent, T., et al.: Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In: ICLR (2024)
6. He, X., Tian, Y., Sun, Y., et al.: G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In: NeurIPS (2024)
7. Hu, E.J., Shen, Y., Wallis, P., et al.: Lora: Low-rank adaptation of large language models (2021), https://arxiv.org/abs/2106.09685
8. Hu, W., Fey, M., Zitnik, M., et al.: Open graph benchmark: Datasets for machine learning on graphs. NeurIPS (2020)
9. Huang, J., Zhang, X., Mei, Q., Ma, J.: Can llms effectively leverage graph structural information through prompts, and why? (2024), https://arxiv.org/abs/2309.16595
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
11. Li, G., Müller, M., Ghanem, B., et al.: Training graph neural networks with 1000 layers. In: ICML (2021)
12. Liu, Z., He, X., Tian, Y., et al.: Can we soft prompt llms for graph learning tasks? In: WWW (2024)
13. Lott, J., Villani, C.: Ricci curvature for metric-measure spaces via optimal transport. Annals of Mathematics (2009)
14. Ni, C.C., Lin, Y.Y., Gao, J., et al.: Network alignment by discrete ollivier-ricci flow. In: International symposium on graph drawing and network visualization (2018)
15. OpenAI: Gpt-4 technical report (2024), https://arxiv.org/abs/2303.08774
16. Sen, P., Namata, G., Bilgic, M., et al.: Collective classification in network data. AI magazine (2008)
17. Tang, J., Yang, Y., Wei, W., et al.: Graphgpt: Graph instruction tuning for large language models (2023)
18. Topping, J., Di Giovanni, F., Chamberlain, B.P., et al.: Understanding over-squashing and bottlenecks on graphs via curvature. arXiv preprint arXiv:2111.14522 (2021)
19. Veličković, P., Cucurull, G., Casanova, A., et al.: Graph attention networks. In: ICLR (2018)
20. Wang, H., Feng, S., He, T., et al.: Can language models solve graph problems in natural language? (2024), https://arxiv.org/abs/2305.10037
21. Ye, Z., Liu, K.S., Ma, T., et al.: Curvature graph network. In: ICLR (2019)