# LLM-GMP: Large Language Model-Based Message Passing for Zero-Shot Learning on Graphs

Md Athikul Islam
mdathikulislam@u.boisestate.edu
Boise State University
Boise, Idaho, USA

Edoardo Serra
edoardoserra@boisestate.edu
Boise State University
Boise, Idaho, USA

## Abstract

Graph-structured data is ubiquitous across scientific and industrial domains, making tasks such as node classification, edge prediction, and graph classification fundamental in modern machine learning. Graph Neural Networks (GNNs) have emerged as the dominant framework for these tasks, leveraging message passing algorithms to propagate information across nodes and learn expressive representations. However, performing zero-shot learning on graphs—where the model must generalize to unseen tasks or labels without additional training—remains highly challenging due to the structural complexity and relational dependencies within graphs.

Recent efforts have explored using Large Language Models (LLMs) for zero-shot reasoning on graphs by converting graph structures into textual descriptions. While promising, these methods face significant limitations due to the restricted context window of LLMs and the risk of hallucinations, especially when processing dense or large-scale graphs.

In this paper, we propose Large Language Model Graph Message Passing (LLM-GMP), a novel framework designed to address the zero-shot learning problem on graphs. Our method combines the scalability of message passing with the reasoning capabilities of LLMs: rather than exchanging vector embeddings as in traditional GNNs, nodes exchange task-aware textual messages, enabling the LLM to explore the graph level by level in a structured, interpretable manner tailored to the downstream task.

By aligning graph exploration with the LLM's strengths in language-based inference, our approach achieves strong zero-shot performance across a range of graph-based tasks, demonstrating the potential of LLM-driven message passing as a powerful alternative to standard graph representation learning methods.

## CCS Concepts

• **Computing methodologies → Machine learning approaches**; **Natural language processing**.

## Keywords

Large Language Models (LLMs), Graph Neural Networks (GNNs), Zero-Shot Learning, Message Passing



*KDD, MLoG-GenAI, Toronto, ON, Canada*



ACM ISBN 978-1-4503-XXXX-X/2025/08
https://doi.org/XXXXXXX.XXXXXXX

**ACM Reference Format:**
Md Athikul Islam and Edoardo Serra. 2025. LLM-GMP: Large Language Model-Based Message Passing for Zero-Shot Learning on Graphs. In *Proceedings of KDD 2025 Workshop on Machine Learning on Graphs in the Era of Generative Artificial Intelligence (KDD, MLoG-GenAI).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 Introduction

Graphs offer a natural and powerful way to represent data with relational and structural dependencies, making them essential in domains such as social network analysis, biological systems, knowledge graphs, and recommendation engines [3, 15, 23]. The advent of GNNs has significantly advanced the state of the art in learning on graph-structured data [17, 25]. By employing message-passing mechanisms that iteratively aggregate information from neighboring nodes, GNNs enable expressive and scalable representations for a variety of tasks, including node classification, link prediction, and graph classification [25, 28].

However, despite these advances, zero-shot learning (ZSL) on graphs remains a persistent challenge [16]. In the zero-shot setting, models are required to generalize to previously unseen tasks or label spaces without additional labeled training data [1]. Traditional GNNs are ill-equipped for this scenario: they typically rely on supervised training tailored to specific tasks and lack the adaptability to handle new objectives without retraining or architectural modifications.

In parallel, LLMs have recently demonstrated impressive zero-shot generalization across a wide range of NLP and reasoning tasks, thanks to pretraining on massive and diverse text corpora [10, 21]. Inspired by this capability, researchers have begun exploring the use of LLMs for graph problems, primarily by translating graph structures into textual formats that LLMs can process [2, 21, 22]. While this approach has shown promise—especially in zero-shot contexts—it is fundamentally limited. Flattening entire graphs into text often exceeds the LLM's context window, particularly for large or densely connected graphs, and can introduce semantic ambiguity or information loss. Moreover, these methods are prone to hallucinations, where the model generates outputs that deviate from the underlying graph structure, undermining reliability in critical applications.

To address these limitations, we propose Large Language Model Graph Message Passing (LLM-GMP), a novel framework that reimagines graph processing through the lens of zero-shot language-based reasoning. Rather than representing the graph as a static text block, LLM-GMP simulates message passing in the language domain: nodes exchange task-aware, interpretable text messages generated by an LLM. These messages evolve over multiple rounds of interaction, allowing information to propagate across the graph in a structured, hierarchical manner. Crucially, the LLM is aware of the downstream task throughout this process, enabling it to synthesize and interpret contextual information effectively.

More concretely, LLM-GMP defines a message passing algorithm in which each node iteratively aggregates textual information received from its neighbors, with the LLM generating messages tailored to the specific zero-shot learning task. Over successive iterations, the LLM refines each node's understanding of its local and global context, culminating in task-specific reasoning (e.g., classification or prediction) based on the final aggregated messages. This paradigm harnesses the complementary strengths of message passing and LLM-based inference, achieving both interpretability and flexibility without retraining.

To the best of our knowledge, this is the first framework to define such a paradigm, which can be viewed as a form of agentic AI, where each node acts as a lightweight reasoning agent within a collaborative, task-driven system. Preliminary experiments show that LLM-GMP offers strong performance across a range of zero-shot graph learning tasks, highlighting its potential as a robust alternative to traditional graph representation learning approaches.

## 2 Related Work

This section reviews two major research directions relevant to our work: traditional GNNs and the emerging field of LLMs operating directly on graph-structured data.

### 2.1 Graph Neural Networks (GNNs)

GNNs have emerged as a powerful framework for learning on graph-structured data, enabling the modeling of complex relationships across various domains such as social networks, molecular chemistry, and recommendation systems. Foundational architectures in this field include Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), and Graph Isomorphism Networks (GINs).

GCNs, introduced by Kipf and Welling [9], extend the concept of convolution from grid-structured data to graph-structured data by employing a localized first-order approximation of spectral graph convolutions. This approach efficiently aggregates feature information from a node's immediate neighbors and has demonstrated strong performance in semi-supervised learning tasks.

To address the limitations of uniform neighbor aggregation in GCNs, Veličković et al. [20] proposed GATs, which incorporate attention mechanisms into the aggregation process. By assigning learnable weights to neighboring nodes, GATs enable nodes to focus on the most relevant parts of their neighborhood, enhancing the model's capacity to capture complex patterns in the data.

Xu et al. [26] introduced GINs to explore the expressive power of GNNs in distinguishing graph structures. GINs employ a sum aggregation function, which, under certain conditions, is as powerful as the Weisfeiler-Lehman graph isomorphism test in distinguishing non-isomorphic graphs. This theoretical foundation allows GINs to achieve superior performance in graph classification tasks.

Comprehensive surveys by Wu et al. [24] and Zhou et al. [29] provide extensive overviews of GNN models, methodologies, and applications, highlighting their versatility and growing impact across various research and application domains.

Despite their success, traditional GNN architectures are not inherently equipped for zero-shot learning tasks.

### 2.2 LLMs Operating Directly on Graphs

The integration of large language models (LLMs) into graph processing has gained considerable traction, showcasing their capacity to reason over complex graph-structured data. Li et al. [13] investigated the structural analysis abilities of LLMs, introducing specialized benchmarks and datasets to support rigorous evaluation. Fan et al. [4] demonstrated that LLMs can effectively augment traditional GNNs, enhancing their representational power and predictive performance. Tang et al. [19] proposed GraphGPT, an instruction-tuned framework that fuses LLMs with graph knowledge, enabling robust generalization across diverse graph datasets.

A parallel line of work focuses on the integration of LLMs with knowledge graphs (KGs). Ibrahim et al. [7] categorized the landscape into three main paradigms: KG-augmented LLMs, LLM-augmented KGs, and hybrid frameworks that combine both methodologies.

Community-driven initiatives, such as the "Awesome-Graph-LLM" repository [6], have emerged to catalog ongoing research and tools at the intersection of graphs and LLMs. Benchmarking platforms like GraphEval36K [18]—which includes 40 graph-related coding tasks and 36,900 test cases—provide comprehensive assessments of LLMs' reasoning capabilities in graph domains. Similarly, GPT4Graph [5] evaluates how well LLMs understand and manipulate graph-structured data, shedding light on their strengths and limitations.

Notably, the most effective zero-shot approaches for graph tasks are those that translate graph structures into textual representations, making them directly accessible to LLMs. However, these methods are constrained by the limited context window of current models and are susceptible to hallucinations, which may affect reliability in critical applications.

## 3 Methodology

The methodology comprises LLM-Guided Message Passing, incorporating Batch-wise Aggregation and tailored LLM Prompt Design to enable effective graph-level reasoning.

### 3.1 LLM-Guided Message Passing with Batch-wise Aggregation

We formulate our task as a *node classification* problem, where the goal is to predict the class label of each document node in a graph. Inspired by the *message passing* paradigm in GNNs, we reimagine node communication using LLMs, which offer semantically rich and adaptive reasoning capabilities. Instead of applying static aggregation functions across all neighbors, we introduce a batch-wise strategy that decomposes message passing into iterative, linguistically grounded updates.

*3.1.1 Batch-wise Aggregation Strategy.* Conventional GNNs aggregate information from all neighboring nodes simultaneously, typically using mathematical operations such as summation, averaging, or attention mechanisms. While effective for numerical features, such approaches often struggle to capture nuanced semantic interactions when applied to textual graphs. To address this limitation, we propose a batch-wise aggregation strategy guided by an LLM, in which each node's representation is refined iteratively by interacting with small, contextually focused subsets of its neighbors.

Let $v_i$ be a node in the document graph with neighborhood $\mathcal{N}(i) = \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$. Rather than aggregating information from the entire neighborhood in a single pass, we partition $\mathcal{N}(i)$ into consecutive mini-batches of size $b$. At each aggregation step $t$, a batch $B_t \subseteq \mathcal{N}(i)$ is selected, and the LLM is prompted with three components: the original text of $v_i$, the current state $h_i^{(t-1)}$, and the texts of neighbors in $B_t$. The inclusion of the original text at every iteration ensures that the node's core semantic identity is preserved across updates, serving as a stable anchor throughout the aggregation process.

Formally, the updated representation at iteration $t$ is computed as

$$h_i^{(t)} = \text{LLM}\left(\text{Prompt}(h_i^{(t-1)}, \text{OriginalText}(v_i), \{h_j^{(t-1)} \mid v_j \in B_t\})\right),$$

where $\text{Prompt}(\cdot)$ constructs a task-specific natural language prompt for the LLM. This iterative procedure continues until all neighbors have been aggregated. The final representation $h_i^{(T)}$, after $T = \lceil |\mathcal{N}(i)|/b \rceil$ iterations, is then used for downstream prediction tasks.

The incremental nature of this strategy offers several advantages. First, it reduces prompt complexity and token length at each LLM call, allowing the model to focus on a small, contextually coherent subset of the graph. Second, it introduces a degree of semantic modularity: each batch-specific aggregation contributes interpretable and traceable updates to the node's evolving representation. Third, by maintaining access to both the original and intermediate node states, the LLM can reason over a richer contextual spectrum, balancing local fidelity and neighbor influence in a structured manner.

The full procedure is outlined in Algorithm 1, which details the initialization of node states, batch scheduling, prompt construction, and representation refinement via LLM-based aggregation. This algorithmic formulation ensures clarity, reproducibility, and seamless integration with transformer-based language models in graph-centric tasks.

## 3.2 LLM Prompt Design

The effectiveness of our message passing framework depends critically on the design of natural language prompts used to guide the LLM. We construct two distinct prompts: one for intermediate message aggregation steps and one for final node classification. These prompts are designed to be informative, modular, and context-aware, allowing the LLM to process localized information while remaining grounded in the global task objective.

Unlike classical GNNs, where feature aggregation is handled numerically, our approach relies on carefully constructed text instructions to stimulate semantic reasoning in the LLM. To this end, each prompt explicitly encodes the task type, iteration context, and relevant input segments, such as the original document text, the current semantic state, and the texts of neighboring nodes. This structure enables the LLM to both preserve semantic grounding and extract contextual signals from the neighborhood.

*3.2.1 Prompt for Message Passing Aggregation.* The message passing prompt is issued iteratively during the batch-wise aggregation process. It instructs the LLM to enhance the node's semantic representation using its own textual content and a small batch of neighboring documents. The prompt is formatted as follows:

**LLM Aggregation Prompt (per iteration)**

**You are assisting with a document classification task on a graph. Each node is a document. You are refining the representation of one document based on its own content and a small batch of neighboring documents.**

**Task:** Enhance the representation of the target document using its own content and its neighbors' content. This enriched representation will be passed to the next iteration and used for final classification.

**Batch Information:** You are currently processing batch #{BATCH_ID} out of {TOTAL_BATCHES}.

**Original Document (always include):**
{ORIGINAL_TEXT}

**Current Document Representation (from previous steps):**
{CURRENT_STATE}

**Neighbor Documents in this batch:**
{NEIGHBOR_TEXT_1}
{NEIGHBOR_TEXT_2}

**Output:** Return a semantically enriched version of the document that integrates both the original text and informative content from the neighbors, suitable for further message passing and eventual classification.

*3.2.2 Prompt for Node Classification.* Once message passing has concluded and the final representation of each node is computed, we issue a classification prompt to the LLM (or optionally a lightweight classifier). This prompt provides the LLM with the enriched semantic representation of the node and instructs it to output a predicted class label:

**LLM Node Classification Prompt**

**You are performing node classification in a graph of documents. Each document belongs to a topic category.**

**Task:** Based on the enriched representation of the document (after message passing), predict the most likely topic.

**Available Classes:** ["sci.space", "rec.sport.hockey", "talk.politics.mideast", ..., "misc.forsale"]

**Final Representation of Document:**
{ENRICHED_DOCUMENT_TEXT}

**Output:** Return only the predicted class label for this document.

**Algorithm 1** LLM-Guided Message Passing with Batch-wise Aggregation

**Require:** Graph $G = (V, E)$, LLM model LLM, batch size $b$, number of message steps $T$

**for all** node $v_i \in V$ **do**
  $h_i^{(0)} \leftarrow \text{OriginalText}(v_i)$
  $N_i \leftarrow \text{ShuffleNeighbors}(\mathcal{N}(i))$
  **for** each batch $B_t = \{v_{j_1}, \ldots, v_{j_b}\} \subseteq N_i$ (non-overlapping) **do**
    $\text{prompt}_t \leftarrow \text{ConstructPrompt}(v_i, h_i^{(t-1)}, \text{OriginalText}(v_i), \{h_j^{(t-1)} \mid v_j \in B_t\})$
    $h_i^{(t)} \leftarrow \text{LLM}(\text{prompt}_t)$
  **end for**
  $\hat{y}_i \leftarrow \text{LLM or Classifier}(h_i^{(T)})$
**end for**
**return** Predicted labels $\{\hat{y}_i\}_{i \in V}$

*3.2.3 Prompt Awareness and Semantic Traceability.* Each prompt is designed to explicitly convey the current step of computation and the role of the output. In the case of message passing, the prompt informs the LLM that the enriched representation will be passed to subsequent iterations and ultimately used for classification, encouraging the model to preserve both semantic fidelity and task relevance. The use of batch indicators (e.g., {BATCH_ID}) further helps disambiguate iterative stages in the aggregation process. Meanwhile, the classification prompt omits structural information, focusing instead on prediction fidelity from the final node embedding.

Together, these prompt designs enable modular, interpretable, and semantically grounded reasoning over graph-structured textual data, forming the linguistic backbone of our LLM-guided message passing framework.

## 4 Experiments

We organize our experiments into four components: implementation settings, baselines, datasets, and preliminary results.

### 4.1 Implementation Settings

We use an NVIDIA GeForce RTX 4090 GPU for our experiments. For all LLM-based operations, we utilize the LLaMA-3.1 model accessed via the open-source library Ollama[1]. To support graph-based reasoning, we construct a $k$-nearest neighbors (k-NN) document graph, where each node represents a document and is initialized with an LLM-derived embedding. Edges are formed by connecting each node to its top-$k$ most semantically similar neighbors based on cosine similarity of the embeddings.

### 4.2 Baselines

We compare our Large Language Model Graph Message Passing against one CNN-based model, one RNN-based model, and one GNN-based baselines.

- **CNN**: A convolutional neural network model for text classification based on the architecture proposed by [8]. We evaluate two settings: CNN-rand, which uses randomly initialized word embeddings, and CNN-non-static, which incorporates pre-trained embeddings that are fine-tuned during training.
- **LSTM**: A long short-term memory network adapted from [14], where the final hidden state represents the entire input sequence. We assess both randomly initialized and pre-trained word embedding variants.
- **GCN**: A graph convolutional network that operates on a heterogeneous document-word graph, learning document-level representations by propagating signals through graph structure, as introduced in [27].

### 4.3 Datasets

We evaluate our approach on two widely used benchmark datasets: 8 Newsgroups (8NG) [11] and R8 [12]. The 8NG dataset is a subset of the original 20 Newsgroups collection, containing 18,846 documents filtered to include only 8 categories, making it suitable for multi-class text classification tasks. The R8 dataset, a subset of the Reuters-21578 corpus, includes documents from 8 categories, with 5,485 samples designated for training and 2,189 for testing. A summary of the key statistics for both datasets is provided in Table 1.

**Table 1: Statistics of Datasets.**

| Dataset | Training Set | Test Set | Average Length | # of Classes |
|---|---|---|---|---|
| 8NG | 2,857 | 894 | 221 | 8 |
| R8 | 5,485 | 2,189 | 66 | 8 |

### 4.4 Preliminary Results

Table 2 reports the classification accuracy on the 8NG and R8 datasets across several baseline models and our proposed approach. The results are divided into two groups: *Supervised Category* and *Unsupervised Category*.

As expected, supervised models such as CNN, LSTM, and GCN achieve strong performance, with GCN performing best overall—achieving 90.27% on 8NG and 98.36% on R8. These models benefit from access to labeled data during training, which enables them to learn task-specific representations effectively. In contrast, our proposed **LLM-GMP** is trained in a completely *unsupervised* setting—without any access to class labels. While the accuracy is lower (76.06% on 8NG and 79.17% on R8), the model demonstrates a meaningful ability to separate categories using only semantic information derived from LLM-guided message passing on document graphs. This highlights the potential of our method in low-resource

[1]https://ollama.com/

or label-scarce scenarios, where collecting annotated data is costly or impractical.

Overall, these results show that our LLM-based approach provides a promising direction for unsupervised text classification, particularly in applications where labeled data is unavailable or limited.

**Table 2: Preliminary classification accuracy (%) of LLM-GMP on 8NG and R8 datasets.**

| Model / Dataset | 8NG | R8 |
|---|---|---|
| *Supervised Category* | | |
| CNN-rand | 78.86 | 96.53 |
| LSTM | 69.69 | 93.10 |
| GCN | 90.27 | 98.36 |
| *Unsupervised Category* | | |
| LLM-GMP | 76.06 | 79.17 |

## 5 Conclusion

We presented **Large Language Model Graph Message Passing** (LLM-GMP), a novel framework for zero-shot graph learning that leverages LLMs to perform message passing via task-aware textual communication between nodes. Unlike traditional GNNs, our method enables interpretable and flexible reasoning without task-specific training, while also overcoming limitations specific to LLM-based graph-to-text methods, such as restricted context windows and hallucinations—by structuring inference as iterative message passing.

Preliminary results show that LLM-GMP performs well across various zero-shot tasks, highlighting the potential of integrating language-based inference with structured graph exploration. By treating nodes as lightweight reasoning agents, our approach opens new avenues at the intersection of GNNs, LLMs, and agentic AI.

Importantly, this framework naturally supports parallel and distributed computation, making it a promising direction for scalable graph reasoning. We anticipate this work will spur further research into efficient, large-scale LLM inference systems tailored to structured data.

## References


[1] Jiaoyan Chen, Yuxia Geng, Zhuo Chen, Jeff Z. Pan, Yuan He, Wen Zhang, Ian Horrocks, and Huajun Chen. 2023. Zero-Shot and Few-Shot Learning With Knowledge Graphs: A Comprehensive Survey. *Proc. IEEE* 111, 6 (2023), 653–685. doi:10.1109/JPROC.2023.3279374

[2] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. 2024. Exploring the Potential of Large Language Models (LLMs)in Learning on Graphs. *SIGKDD Explor. Newsl.* 25, 2 (March 2024), 42–61. doi:10.1145/3655103.3655110

[3] Zhe Chen, Yuehan Wang, Bin Zhao, Jing Cheng, Xin Zhao, and Zongtao Duan. 2020. Knowledge Graph Completion: A Review. *IEEE Access* 8 (2020), 192435–192456. doi:10.1109/ACCESS.2020.3030076

[4] Wenqi Fan, Shijie Wang, Jiani Huang, Zhikai Chen, Yu Song, Wenzhuo Tang, Haitao Mao, Hui Liu, Xiaorui Liu, Dawei Yin, et al. 2024. Graph machine learning in the era of large language models (llms). *arXiv preprint arXiv:2404.14928* (2024).

[5] Jiayan Guo, Lun Du, Hengyu Liu, Mengyu Zhou, Xinyi He, and Shi Han. 2023. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066* (2023).

[6] Xiaoxin He. 2024. Awesome-Graph-LLM: A collection of AWESOME things about Graph-Related LLMs. https://github.com/XiaoxinHe/Awesome-Graph-LLM Accessed: 2025-03-27.

[7] Nourhan Ibrahim, Samar Aboulela, Ahmed Ibrahim, and Rasha Kashef. 2024. A survey on augmenting knowledge graphs (KGs) with large language models (LLMs): models, evaluation metrics, benchmarks, and challenges. *Discover Artificial Intelligence* 4, 1 (2024), 76.

[8] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1746–1751.

[9] Thomas N Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1609.02907

[10] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 22199–22213. https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf

[11] Ken Lang. 1995. Newsweeder: Learning to filter netnews. Carnegie Mellon University, School of Computer Science, Technical Report. Dataset available at http://qwone.com/~jason/20Newsgroups/.

[12] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. RCV1: A new benchmark collection for text categorization research. In *Journal of Machine Learning Research*, Vol. 5. 361–397.

[13] Xin Li, Weize Chen, Qizhi Chu, Haopeng Li, Zhaojun Sun, Ran Li, Chen Qian, Yiwei Wei, Chuan Shi, Zhiyuan Liu, et al. 2024. Can large language models analyze graphs like professionals? a benchmark, datasets and models. *Advances in Neural Information Processing Systems* 37 (2024), 141045–141070.

[14] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*. 2873–2879.

[15] Sameh K Mohamed, Aayah Nounu, and Vít Nováček. 2020. Biological applications of knowledge graph embedding models. *Briefings in Bioinformatics* 22, 2 (02 2020), 1679–1693. doi:10.1093/bib/bbaa012 arXiv:https://academic.oup.com/bib/article-pdf/22/2/1679/36654277/bbaa012.pdf

[16] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. 2021. Learning Graph Embeddings for Compositional Zero-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 953–962.

[17] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* 20, 1 (2009), 61–80. doi:10.1109/TNN.2008.2005605

[18] Kun et al. Shen. 2024. GraphEval36K: A Benchmark for Evaluating Graph Understanding in LLMs. *Proceedings of CVPR* (2024).

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

[20] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1710.10903

[21] Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. 2024. LLMs as Zero-shot Graph Learners: Alignment of GNN Representations with LLM Token Embeddings. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 5950–5973. https://proceedings.neurips.cc/paper_files/paper/2024/file/0b77d3a82b59e9d9899370b378087faf-Paper-Conference.pdf

[22] Shijie Wang, Jiani Huang, Zhikai Chen, Yu Song, Wenzhuo Tang, Haitao Mao, Wenqi Fan, Hui Liu, Xiaorui Liu, Dawei Yin, and Qing Li. 2025. Graph Machine Learning in the Era of Large Language Models (LLMs). *ACM Trans. Intell. Syst. Technol.* (May 2025). doi:10.1145/3732786 Just Accepted.

[23] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph Neural Networks in Recommender Systems: A Survey. *ACM Comput. Surv.* 55, 5, Article 97 (Dec. 2022), 37 pages. doi:10.1145/3535101

[24] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. 2020. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2020), 4–24.

[25] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2021), 4–24. doi:10.1109/TNNLS.2020.2978386

[26] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1810.00826

[27] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7370–7377.

[28] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks:

A review of methods and applications. *AI Open* 1 (2020), 57–81. doi:10.1016/j.aiopen.2021.01.001
[29] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph Neural Networks: A Review of Methods and Applications. *AI Open* 1 (2020), 57–81.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009