Neural Networks 191 (2025) 107777

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

Democratizing large language model-based graph data augmentation via latent knowledge graphs

Yushi Feng[1], Tsai Hor Chan[1], Guosheng Yin, Lequan Yu *

Department of Statistics and Actuarial Science, School of Computing and Data Science, The University of Hong Kong, Hong Kong Special Administrative Region of China

ARTICLE INFO

Dataset link: https://github.com/HKU-MedAI/DemoGraph, https://physionet.org/content/mimiciii/1.4/, https://physionet.org/content/?topic=mimic-iv

Keywords:
Large language models
Graph representation learning
Knowledge graphs
Medical informatics
Data augmentation

ABSTRACT

Data augmentation is necessary for graph representation learning due to the scarcity and noise present in graph data. Most of the existing augmentation methods overlook the context information inherited from the dataset as they rely solely on the graph structure for augmentation. Despite the success of some large language model-based (LLM) graph learning methods, they are mostly white-box which require access to the weights or latent features from the open-access LLMs, making them difficult to be democratized for everyone as the most advanced LLMs are often closed-source for commercial considerations. To overcome these limitations, we propose a black-box context-driven graph data augmentation approach, with the guidance of LLMs — **DemoGraph**. Leveraging the text prompt as context-related information, we task the LLM with generating knowledge graphs (KGs), which allow us to capture the structural interactions from the text outputs. We then design a dynamic merging schema to stochastically integrate the LLM-generated KGs into the original graph during training. To control the sparsity of the augmented graph, we further devise a granularity-aware prompting strategy and an instruction fine-tuning module, which seamlessly generates text prompts according to different granularity levels of the dataset. Extensive experiments on various graph learning tasks validate the effectiveness of our method over existing graph data augmentation methods. Notably, our approach excels in scenarios involving electronic health records (EHRs), which validates its maximal utilization of contextual knowledge, leading to enhanced predictive performance and interpretability.

1. Introduction

Graph representation learning has received increasing attention in recent years. It achieves great success in solving tasks where relational features are important, such as recommendation systems (Cai, Huang, Xia, & Ren, 2023; Shi, Hu, Zhao, & Philip, 2018), citation networks (Hu, Fey, et al., 2020), and medical records analysis (Choi, Xiao, Stewart, & Sun, 2018; Ma et al., 2018). However, the scarcity and noise present in graph data pose great challenges for effective graph learning, necessitating the development of graph data augmentation algorithms.

Existing graph data augmentation methods focus on graph structures for data augmentation, such as randomly dropping nodes or edges, adding Gaussian noise to the node or edge attributes, or applying graph-based transformations such as sub-sampling and node permutation. While these methods have demonstrated some successes in graph representation learning scenarios, they do not consider the *context* or *attributes* associated with the graph data. This prompts some recent works (He et al., 2023; Huang, Zeng, Wu, & Lü, 2024; Jiang, Xiao, Cross, & Sun, 2023; Tang et al., 2023; Wei et al., 2024; West et al., 2021; Zhang et al., 2022) which leverage LLM for graph representation learning. Despite their success, they are mostly white-box which require access to the weights or latent features from the open-access LLMs. While numerous open-source LLMs exist, the most advanced models are often closed-source for commercial reasons, posing significant challenges to democratize these methods for broader use. As a result, the resulting augmented graph becomes less identifiable due to a lack of contextual guidance. Furthermore, most of these augmentation methods leverage in-domain knowledge under a close-world setting, which does not borrow the vast repositories of knowledge in the open world. Additionally, the sparsity of the augmented graph is not well studied, although some methods, such as DropEdge, attempt to sparsify the graph for augmentation. Without proper sparsity control, the augmented graph would be over-sparsified and likely reduced to trivial graphs (i.e., uninformative graphs). These limitations pop the necessity of developing a new graph data augmenter under open-world

* Corresponding author.
E-mail address: lqyu@hku.hk (L. Yu).
[1] Equal contributions.

https://doi.org/10.1016/j.neunet.2025.107777
Received 13 November 2024; Received in revised form 16 April 2025; Accepted 16 June 2025
Available online 11 July 2025

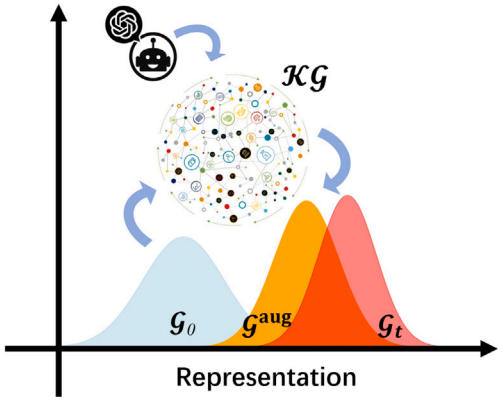

Fig. 1. Schematic illustration of the feature distribution of original graph $\mathcal{G}_0$ from observations and $\mathcal{G}^{aug}$, which represents the augmented graph for $\mathcal{G}_0$ after merging the context knowledge in terms of $\mathcal{KG}$. After performing graph data augmentation with LLM-guided DemoGraph, $\mathcal{G}^{aug}$ is closer to the true representation $\mathcal{G}_t$.

settings with proper sparsity control, such that the augmented graph can be closer to the true data distribution (see Fig. 1).

In light of the vast development of large language models (LLMs), we propose a novel framework, namely **DemoGraph**, to perform contextual graph data augmentation with a generative pretrained LLM. Our contributions can be summarized as (1) We introduce a black-box method that leverages extensive knowledge from LLM to perform graph data augmentation without access to model weights or source codes. This is particularly realistic when most LLMs are provided in close-source commercial APIs, enabling the democratization of LLM-based methods. We adopt latent KGs to capture the structural interactions from the text outputs, as well as a compatible data structure for graph data. (2) We design a dynamic merging strategy to stochastically integrate the LLM-generated KGs into the raw graph data during the network training, which guides the optimization trajectory with contextual knowledge. (3) To tackle the sparsity induced by generated KGs, we design a granularity-aware prompting strategy to control the sparsity while maximizing the utility of domain knowledge. Also, we leverage a sequential prompting with instruction fine-tuning strategy to incentivize the LLM to generate the most relevant concepts to the context, and hence high-quality KGs. (4) Extensive experiments on various graph learning tasks validate the effectiveness of our method over existing graph data augmentation methods. (5) Our method demonstrates high scalability across datasets ranging from small to large-scale, consistently delivering satisfactory performance. Notably, our approach excels in scenarios involving electronic health records (EHRs), where our method maximizes the utilization of contextual information, and leads to enhanced predictive performance and interpretability.

2. Related works

Graph Neural Networks (GNNs). GNNs are gaining significant success in many problem domains (Chan, Wong, Shen, & Yin, 2023; Hu, Dong, et al., 2020; Kojima et al., 2020; Liu, Li, Peng, He, & Philip, 2020; Simonovsky & Komodakis, 2018; Wu, Ren, Li, & Leskovec, 2020). They learn node representation by aggregating information from the neighboring nodes on the graph topology. Most of the existing GNN architectures are on homogeneous graphs (Veličković, Cucurull, Casanova, Romero, Lio, & Bengio, 2017; Welling & Kipf, 2016; Xu, Hu, Leskovec, & Jegelka, 2018; Yun, Jeong, Kim, Kang, & Kim, 2019). There are also GNN architectures that operate on heterogeneous graphs to learn its enriched structural information and complex relations (Hu, Dong, et al., 2020; Huang, Xu, & Wang, 2020; Schlichtkrull et al., 2018; Wang, Ji, et al., 2019; Yang, Song, Jin, & Du, 2020). However, due to limited samples, it is difficult to approximate the true data distribution, especially in the graph domain. Hence, an effective graph data augmentation algorithm is needed to boost the performance of GNNs.

Graph Data Augmentation. Graph data augmentation (GDA) aims to enhance the utility of the input graph data and produce graph samples close to the true data distribution to alleviate the finite sample bias (Ding, Xu, Tong, & Liu, 2022). Most of the existing works focus on perturbating the graph structures or node features/labels to achieve augmentation, such as node dropping (Feng et al., 2020), edge perturbation (Rong, Huang, Xu, & Huang, 2019; Veličković, Fedus, Hamilton, Liò, Bengio, & Hjelm, 2018), graph rewriting (Franceschi, Niepert, Pontil, & He, 2019; Wang et al., 2020; Yang et al., 2019), graph sampling (Hamilton, Ying, & Leskovec, 2017a, 2017b; Qiu et al., 2020), graph diffusion (Park et al., 2021; Qiu et al., 2020; Topping, Di Giovanni, Chamberlain, Dong, & Bronstein, 2021; Zheng et al., 2020) or pseudo-labeling (Zhang, Cisse, Dauphin, & Lopez-Paz, 2017). There are also works that adopt a learnable graph data augmenter and design specific losses for training (Li, Han, & Wu, 2018; Liu, Ying, et al., 2022; Park, Shim, & Yang, 2022; Suresh, Li, Hao, & Neville, 2021; Wu et al., 2020; You, Chen, Wang, et al., 2020). However, these methods mainly focus on the graph structures without considering the contextual information or introducing open-world knowledge.

Recent works He et al. (2023), Huang et al. (2024), Jiang et al. (2023), Tang et al. (2024), Wei et al. (2024), West et al. (2021), Zhang et al. (2022), Zhao, Qu, et al. (2023) on LLM-based GDA have achieved promising improvements. However, current LLM-based methods are mostly white-box which require access to the weights or latent features from the LLMs. It is computationally inefficient and impractical, as SOTA LLMs are costly for large-scale experiments and often closed-source. Additionally, they often distinctly require enriched contextual information for specific tasks (e.g. detailed abstract for academic publications (Chen, Feng, He, Deng, Pu, & Li, 2025; He et al., 2023), clinical reports for medical tasks (Jiang et al., 2023) or detailed text annotation at single granularity (Zhao, Qu, et al., 2023)), hindering their generalizability and performance in broader graph learning scenarios. Moreover, these methods mostly focus on node-level context and neglect the higher-order graph structures. Hence, a black-box LLM-based GDA framework with awareness of higher-level graph structure is needed to address these limitations. Moreover, these methods mostly focus on node-level context and neglect the higher-order graph structures. Hence, a black-box LLM-based GDA framework with awareness of higher-level graph structure is needed to address these limitations.

Graph Learning in Healthcare. Knowledge distillation from massive EHRs has been a popular topic in healthcare informatics. To address the longitudinal features in the EHR data, several early works (Ma et al., 2017; Ma, Gao, et al., 2020; Ma, Zhang, et al., 2020) attempted to learn the EHR features using recurrent neural networks. Since the EHR data represent relational information between entities (e.g., patients make visits), graphical models turn out to be an ideal approach for representing the EHR data (Choi, Bahadori, Song, Stewart, & Sun, 2017; Choi et al., 2018). GRAM (Choi et al., 2017) is a well-known method that learns robust medical code representations by adopting a graph-based attention mechanism. However, a critical gap remains in these methods: they do not fully incorporate the rich contextual information available in EHR data (Fiol et al., 2013; Hsu, Taira, El-Saden, Kangarloo, & Bui, 2012). This oversight can lead to a lack of nuanced understanding of patient data, impacting the accuracy and applicability of the derived insights (Evans, 2016). Furthermore, there is a notable absence of effective regularization mechanisms for adjusting to the inherent noise in EHR data, which is cluttered with irrelevant or redundant information.

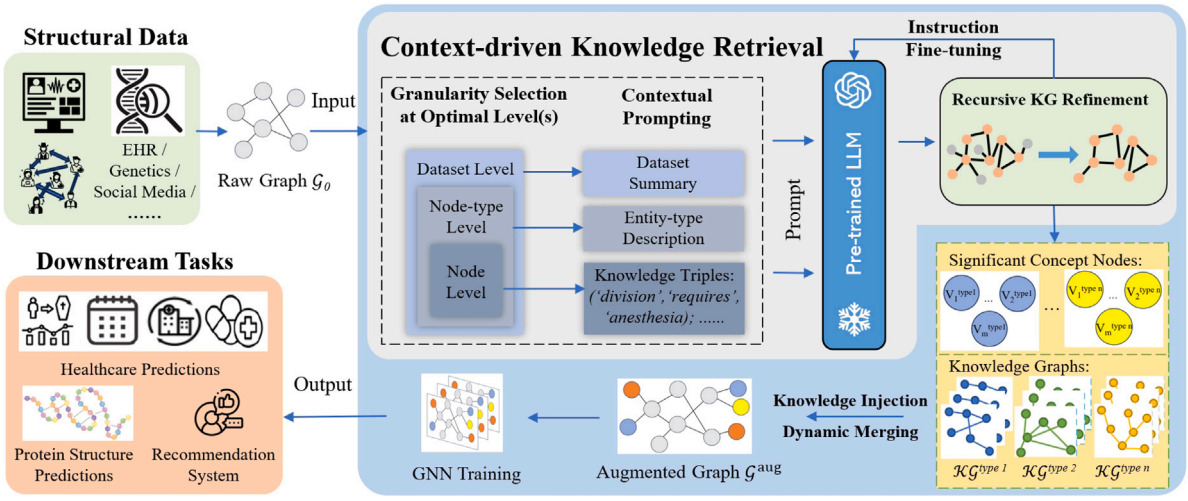

Fig. 2. Overview of our proposed DemoGraph framework. Given a dataset, we first construct a graph $\mathcal{G}_0$ to highlight the relational information, and then perform context-driven knowledge retrieval by utilizing the original dataset and a frozen generative pre-trained LLM. We conduct contextual, adaptive, sparsity-controllable and granularity-aware prompt learning on the LLM, thus obtaining either concept-specific KGs or important extra concept nodes at different levels after refinement. For the original graph $\mathcal{G}_0$, we perform graph data augmentation with the domain-knowledge injection procedure. We train a GNN model on the augmented graph $\mathcal{G}^{\text{aug}}$, thus our framework is able to handle a wide range of downstream tasks across various domains depending on the original datasets.

3. Preliminaries

Graphs. A graph $\mathcal{G}$ is a collection of vertices $\mathcal{V}$ and edges $\mathcal{E}$, typically represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each edge $e \in \mathcal{E}$ is an ordered or unordered pair of vertices representing the connection between them. In the context of graph neural networks, each vertex v_i is often associated with a feature vector x_i in the feature space $\mathcal{X}$. A knowledge graph (KG) is a specialized type of graph denoted as $\mathcal{KG} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where $\mathcal{R}$ is a set of relation types. A KG can be constructed from a set of triples $\mathcal{T} = \{(h_i, r_i, t_i)\}_{i=1}^{|\mathcal{T}|}$ where h_i, t_i, and r_i are the ith head and tail nodes respectively, and r_i is the relation type for the ith triple.

Graph Data Augmentation (GDA). Given $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, GDA aims to derive an augmented graph $\mathcal{G}^{\text{aug}} = (\mathcal{V}^{\text{aug}}, \mathcal{E}^{\text{aug}})$, where $\mathcal{V}^{\text{aug}}$ and $\mathcal{E}^{\text{aug}}$ represent the augmented set of nodes and edges, respectively. The augmentation process should preserve or enhance the inherent structure and properties of $\mathcal{G}$, while facilitating the improved performance of a GNN (denoted as $\mathcal{M}$) on downstream tasks.

4. Methodology

Our proposed framework consists of two main modules: a knowledge graph construction module with leveraging knowledge from LLMs, and a graph data augmentation module with dynamic knowledge injection. Fig. 2 and Algorithm 1 provide an overview of the workflow of our framework.

4.1. Context-driven knowledge retrieval

General Prompting Strategy. The cornerstone of our framework is the construction of KGs using LLMs. The context-aware KGs serve as enriched contextual domain knowledge that augments the original graph $\mathcal{G}_0$ towards the true representation $\mathcal{G}_t$. The KG construction is facilitated through a prompting mechanism that steers the LLM toward generating subgraphs focused on specific concepts. The generation process in general can be formulated as $\mathcal{T} \leftarrow \text{LLM}(\text{prompt})$, where $\mathcal{T} = \{(h_i, r_i, t_i)\}_{i=1}^{|\mathcal{T}|}$ represents the set of triples indicating the relationships between the generated concepts. A knowledge graph $\mathcal{KG}$ can then be constructed from $\mathcal{T}$. We design modularized prompts (with placeholders for the descriptions) that are based on all the available information (e.g., the summary of datasets, task descriptions) of the working graph dataset, such that context knowledge can be maximally utilized. One example of the prompting design on the EHR context is: where the variables as placeholders are inside {} — `{example}` provides an exemplar triple format, `{descriptions}` offers the contextual information, and "`updates:`" prompts the LLM to finish the paragraph. This prompt initially instructs the LLM to identify and generate concept entities $\mathcal{V}^{\mathcal{KG}}$ and their interrelations $\mathcal{E}^{\mathcal{KG}}$ driven by the descriptions (e.g., on the dataset or entity) and oriented to the target tasks. Subsequently, the LLM regularizes these relationships into standardized triple formats. Finally, the above prompt expands this structured information both in width and depth, digging into more meaningful and nested relationships, until a pre-defined number of triples is reached. We also prompt example triples to regularize the output formats of $\mathcal{T}$. This multi-step process ensures that the KG is both information-rich and aligned with domain-specific objectives.

Algorithm 1 The training workflow of our graph data augmentation method.

Input: Original graph $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$ with randomly-initialized node features $\{x_i, \forall i \in \mathcal{V}\}$, granularity level s, number of KGs generated K (per step), ground truth labels y.
Output: Augmented graph $\mathcal{G}^{\text{aug}}$, trained GNN model $\mathcal{M}$.
Initialize $\mathcal{G}^{\text{aug}} = \mathcal{G}_0$
for each epoch **do**
  $\mathcal{V}^{\mathcal{KG}} \leftarrow$ Get concept nodes as augmentation entities,
  $\{\mathcal{KG}\}_{i=1}^{K} \leftarrow$ Load KGs from $\mathcal{V}^{\mathcal{KG}}$,
  $\{\mathcal{KG}\}_{i=1}^{K} \leftarrow$ Perform instruction fine-tuning with customized sparsity control on $\{\mathcal{KG}\}_{i=1}^{K}$,
  $\mathcal{G}^{\text{aug}} \leftarrow \text{merge\_KG}(\{\mathcal{KG}\}_{i=1}^{K}, \mathcal{G}^{\text{aug}})$,
  Update node indices for all node types in $\mathcal{G}^{\text{aug}}$,
  Get prediction from the GNN $\hat{y} = \mathcal{M}(\mathcal{G}^{\text{aug}})$,
  Compute training loss $\mathcal{L}(\hat{y}, y)$,
  Backpropagate $\mathcal{L}$ to $\mathcal{M}$
end for
return Trained GNN $\mathcal{M}$

Notably, this paradigm utilizing placeholders avoids manual prompt customization, thereby reducing human labor costs.

```
Start with the following prompt on a given medical
    concept (such as health condition/treatment
    procedure/drug) and generate an extensive array of
    associated connections based on your domain
    knowledge. These connections should help improve
    prediction tasks in healthcare, e.g. drug
    recommendation, mortality prediction, length of
    stay and readmission prediction.
Format each association as [ENTITY 1, RELATIONSHIP,
    ENTITY 2], ensuring the sequence reflects the
    direction of the relationship. Both ENTITY 1 and
    ENTITY 2 are to be nouns. Elements within [ENTITY 1,
    RELATIONSHIP, ENTITY 2] must be definitive and
    succinct.
Approach in both breadth and depth. Continue expanding [
    ENTITY 1, RELATIONSHIP, ENTITY 2] combinations
    until reaching a total of 100.
{example}
prompt: {descriptions}
updates:
```

Granularity-Aware Prompting for Sparsity Control. Naively utilizing the prompting strategy in the previous section would mostly lead to a sparse KG, where data points are unevenly distributed with many gaps or missing links. Hence, we propose a multi-layer augmentation strategy that determines a granularity level prior to generation, such that the sparsity of the KG can be controlled.

Granularity refers to the data scale of detail in the augmentation process, ranging from coarse-grained dataset level to fine-grained node level information. Based on the availability of information in the working dataset, we define s as the sparsity level parameter (s increases as the data are more fine-grained), and separate the prompting strategy into three granularity levels, $s_0 < s_1 < s_2$, as follows:

- **Dataset-level Augmentation** ($s = s_0$). At the dataset level, our objective is to identify and propagate overarching themes and concepts that are broadly relevant across the dataset. This macro approach involves curating concepts and triples that reflect high-level semantics and dependencies. This is the most fundamental form of our method since dataset-level information is always available.
- **Type-level Augmentation** ($s = s_1$). Another common scenario is that we have node type level information (e.g., class labels in texts for classification). We distill the most salient concepts and relationships pertinent to each class or node type. By doing so, we gain an in-depth understanding of the node categories, fleshing out their characteristics and the interconnections within them. A node-type level prompting example on the Cora dataset (7 classes) is provided in the appendix.
- **Node-level Augmentation** ($s = s_2$). In some scenarios (e.g., EHR datasets), we have the finest information (e.g., text description) on each node (or medical entity). At this juncture, we aim to enrich individual nodes with highly relevant and specific concepts that are crucial for the particular tasks. This targeted augmentation ensures that nodes are imbued with unique attributes that can drive predictive tasks more effectively.

Concept Pruning via Instruction Fine-tuning. Due to the high complexity of given tasks, LLM's one-time retrieval of KGs may contain low-entropy (i.e., uninformative) concepts (e.g., *is*, *dataset*, or *disease*). We thus instruct LLMs to go through a chain-of-thought process to do multi-stage reasoning and self-improve the quality of KGs. Fig. 3 illustrates our concept prompting procedure via instruction fine-tuning. Given the initial generated $\mathcal{KG}$, we refine it by recursively calling

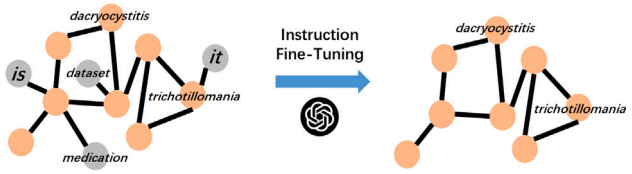

Fig. 3. Concept pruning via instruction fine-tuning, where trivial concepts can be pruned by re-prompting the coarse set of concepts to the LLM.

the LLM and pruning less relevant nodes and edges, while ensuring that a predefined percentage of the concepts are directly derived from the original dataset. A template for this instruction fine-tuning (IFT) process is given below (we use EHR as an illustrative example). After this procedure, a set of important concept nodes $\mathcal{V}^{\mathcal{KG}}$ is then output for triple construction and KG generation.

```
Given the list of triples augmented with MIMIC-III
    dataset, I want to select '{number_of_concepts}'
    most important triples from the list. The importance
     of a triple is based on your knowledge and inference
    on how it will help improve prediction tasks in
    healthcare, e.g. drug recommendation, mortality
    prediction, length of stay, readmission prediction.
     If you think a triple is important, please keep it.
    Otherwise, please remove it. You can also add
    triples from your background knowledge.
triples: {triples}
updates:
```

4.2. Augmentation with generated KGs

Dynamic Graph Merging. The motivation behind designing dynamic merging is to ensure that the GNN learns meaningful information by integrating KGs with raw graphs. This process allows the GNN to explore different ways of connecting the raw graph and the KG at each training step, enabling the model to optimize the graph structure for the specific task at hand.

Specifically, we provide a detailed explanation as follows: First, we perform node and edge selection. For each concept node $v_c \in \mathcal{V}^{\mathcal{KG}}$ in the KG, we select a subset of nodes $\mathcal{V}_0^s = \{z \mid z \in \mathcal{V}_0\}$ from the base graph $\mathcal{G}_0$, where n_c is the predetermined number of edges per concept node. These nodes are then connected to the concept nodes to form an edge set

$$\mathcal{E}^{\text{conn}} = \{(v_c, z) | \forall v_c \in \mathcal{V}^{\mathcal{KG}}, z \in \mathcal{V}_0^s\}.$$

The edge set between the original graph and the augmented graph is determined by the training loss. During training, the set is dynamically selected and updated so that the training loss can be minimized.

The augmented graph $\mathcal{G}^{\text{aug}} = (\mathcal{V}^{\text{aug}}, \mathcal{E}^{\text{aug}})$ is obtained by combining the edge sets and node sets from both the original and augmented graphs: $\mathcal{E}^{\text{aug}} = \mathcal{E}^{\text{conn}} \cup \mathcal{E}_0 \cup \mathcal{E}^{\mathcal{KG}}$ and $\mathcal{V}^{\text{aug}} = \mathcal{V}_0 \cup \mathcal{V}^{\mathcal{KG}}$.

During iterative updates, unlike a one-off merging process, dynamic merging is iterative. In each training epoch, the model updates the KG based on its current state, ensuring the graph data remains dynamic and contextually relevant. This iterative approach helps prevent overfitting and improves the model's generalization to unseen data.

Due to the computation limitations, the number of LLM inferences is limited. Therefore, we precompute $\mathcal{KG}$ offline and merge it with $\mathcal{G}_0$ stochastically during training. Under sufficient computational conditions, the dynamic merging schema allows for online prompting where an up-to-date $\mathcal{KG}$ can be generated after every optimization step. On the other hand, the LLM can also be fine-tuned online with task-specific losses. This allows for more context-related KG generations and hence

Table 1
Performance evaluation of various GNN architectures on graph learning tasks. The PPI dataset is employed for graph classification (assessed via Micro-F1), while the remaining datasets (Cora, Citeseer, and Actor) are used for node classification (evaluated using Micro-F1 and Accuracy). Standard deviations are indicated in parentheses.

GNN Archi.	Augmenter	PPI Micro-F1	Cora Micro-F1	Citeseer Micro-F1	Actor Accuracy	Cora Accuracy	Citeseer Accuracy
Graph SAGE	None	60.0 (2.7)	82.8 (3.6)	69.1 (2.5)	36.7 (1.8)	81.0 (3.3)	70.9 (2.0)
	DropNode (Feng et al., 2020)	61.5 (2.6)	81.4 (3.4)	68.0 (2.4)	36.8 (1.5)	80.6 (3.2)	70.1 (2.7)
	DropEdge (Rong, Huang, Xu, & Huang, 2020)	63.2 (3.1)	81.6 (2.8)	70.4 (2.6)	36.8 (2.9)	80.4 (2.8)	71.2 (3.2)
	RandomwalkPE (Dwivedi, Luu, Laurent, Bengio, & Bresson, 2021)	63.1 (2.7)	82.0 (2.6)	68.0 (1.5)	37.7 (2.7)	81.2 (3.1)	70.8 (2.6)
	LaplacianPE (Dwivedi et al., 2023)	63.5 (3.1)	81.9 (2.1)	69.7 (1.9)	36.7 (2.1)	80.9 (2.2)	70.7 (2.5)
	GraphGPT (Tang et al., 2024)	90.2 (3.0)	82.2 (2.7)	70.5 (2.2)	37.2 (2.5)	81.5 (2.9)	71.0 (2.6)
	DemoGraph (Ours)	**93.6 (2.3)**	**83.3 (2.0)**	**71.7 (1.2)**	**37.9 (1.6)**	**83.3 (1.2)**	**72.6 (2.0)**
GAT	None	97.1 (3.0)	82.0 (4.0)	71.0 (3.6)	30.3 (2.7)	82.1 (4.3)	72.1 (3.7)
	DropNode (Feng et al., 2020)	94.0 (3.4)	80.5 (3.7)	71.2 (3.3)	31.3 (2.2)	80.7 (3.7)	71.9 (3.2)
	DropEdge (Rong et al., 2020)	85.1 (3.0)	79.1 (3.8)	68.8 (3.8)	31.2 (3.0)	78.9 (3.9)	69.1 (3.9)
	RandomWalkPE (Dwivedi et al., 2021)	90.8 (3.6)	81.3 (2.9)	71.2 (3.1)	31.4 (2.5)	81.2 (3.2)	71.9 (3.2)
	LaplacianPE (Dwivedi et al., 2023)	90.7 (2.7)	81.5 (2.5)	71.4 (2.6)	30.9 (2.9)	81.4 (2.4)	71.8 (2.7)
	GraphGPT (Tang et al., 2024)	95.9 (3.3)	82.2 (3.6)	71.9 (3.4)	31.5 (2.9)	81.6 (3.1)	71.6 (3.0)
	DemoGraph (Ours)	**97.2 (3.4)**	**83.6 (3.2)**	**72.4 (2.3)**	**32.2 (2.3)**	**83.6 (2.0)**	**73.1 (2.2)**
GCN	None	53.2 (2.4)	78.4 (3.4)	71.6 (2.7)	29.8 (2.1)	81.0 (2.7)	69.4 (2.0)
	DropNode (Feng et al., 2020)	58.9 (1.9)	79.2 (2.6)	72.2 (1.5)	28.7 (2.5)	78.9 (2.6)	70.5 (2.0)
	DropEdge (Rong et al., 2020)	54.8 (4.1)	82.2 (3.9)	71.5 (2.7)	28.9 (3.4)	82.4 (3.5)	71.3 (3.2)
	RandomWalkPE (Dwivedi et al., 2021)	59.0 (1.6)	80.9 (2.2)	71.8 (2.4)	29.8 (2.9)	80.0 (2.9)	71.6 (2.2)
	LaplacianPE (Dwivedi et al., 2023)	59.3 (1.6)	80.4 (2.1)	71.3 (1.9)	29.6 (2.2)	80.0 (1.9)	71.1 (2.1)
	GraphGPT (Tang et al., 2024)	59.1 (1.8)	82.0 (2.9)	72.1 (2.1)	30.2 (2.6)	81.6 (1.9)	71.8 (2.1)
	DemoGraph (Ours)	**60.3 (1.2)**	**82.7 (2.9)**	73.1 (1.9)	**32.4 (2.3)**	**82.9 (1.0)**	**73.1 (1.1)**
GIN	None	70.3 (2.8)	81.0 (4.1)	70.8 (3.7)	31.9 (2.0)	81.6 (2.0)	70.9 (3.7)
	DropNode (Feng et al., 2020)	75.2 (3.1)	79.1 (4.2)	70.8 (4.1)	32.4 (2.2)	78.5 (4.1)	70.6 (4.0)
	DropEdge (Rong et al., 2020)	78.3 (3.7)	81.8 (3.9)	69.0 (3.8)	32.7 (2.8)	81.8 (4.4)	71.5 (3.9)
	RandomWalkPE (Dwivedi et al., 2021)	76.2 (3.5)	81.1 (3.3)	69.8 (3.6)	33.1 (2.5)	80.9 (2.7)	71.1 (3.8)
	LaplacianPE (Dwivedi et al., 2023)	74.5 (2.9)	80.0 (2.7)	69.9 (3.7)	32.9 (2.4)	81.9 (2.7)	71.4 (3.6)
	GraphGPT (Tang et al., 2024)	78.2 (3.0)	81.6 (4.6)	71.0 (4.1)	33.0 (2.5)	81.9 (4.4)	71.6 (4.1)
	DemoGraph (Ours)	**79.2 (2.8)**	**82.2 (4.9)**	72.2 (4.2)	**34.8 (2.2)**	**82.3(4.5)**	**72.9 (3.9)**

improved data augmentation performance. It also enables the potential for training open-world GNN models.

Training Paradigm. We use GNN to predict the labels with the augmented graph as the input, $\hat{y} = \mathcal{M}(\mathcal{G}^{\text{aug}})$. We benchmark with different choices of $\mathcal{M}$: graph convolutional network (GCN) (Welling & Kipf, 2016), graph attention network (GAT) (Veličković et al., 2017), GraphSAGE (Hamilton et al., 2017a), and graph isomorphism network (GIN) (detailed formulations and descriptions of GNNs in appendix). We compute the loss for backpropagation with the predictive labels. For instance, in a multi-class classification task, we adopt the cross-entropy loss, $L_{\text{ce}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\text{softmax}(z_{i,c}))$, where $y_{i,c}$ is the ground truth label for patient i and class c, N is the number of observations, C is the number of classes, and $z_{i,c}$ is logits obtained from the model.

4.3. Adaptability to other graph datasets

Since EHR contains enriched contextual information that allows for flexible prompting design, we use the EHR dataset to illustrate our prompting strategy. However, our prompting strategy is adaptable to other graph datasets, as the placeholders in the modularized prompts can be replaced by information on the target datasets. We can also incrementally enlarge the KG such that knowledge from the existing domain can be leveraged to the target domain. We employ a highly-adaptive customization strategy that tailors the prompt structure based on the specific dataset in use. This strategy includes understanding the data's content and structure and then adjusting the prompts to ensure the generated KGs are optimally suited for the data in question.

5. Experiments

5.1. Experimental settings

Datasets and Tasks. (1) We perform experiments on **generic** graph benchmarks (Cora, PPI, Actor, and Citeseer), where we benchmark our method on node classification tasks. (2) We validate the scalability of DemoGraph on two **large-scale** datasets — OGBN-products and OGBN-arxiv (Hu, Fey, et al., 2020) against additional LLM-based methods. Table B.11 and B.12 provide a summary of these graph datasets from small to large-scales. (3) Additionally, we highlight an application of our method on a **large-scale EHR** dataset — MIMIC-III (Johnson et al., 2016). It contains a publicly available dataset of 46,520 intensive care unit (ICU) patients over 11 years. We perform four supervised tasks — in-hospital mortality prediction (MORT), readmission prediction (READM), length of stay (LOS) prediction, and drug recommendations (DR), where MORT and READM predictions are approached as binary classification tasks, LOS prediction as a multi-class classification task, and DR as a multi-label classification task. Since the lab events are sparse and introduce heavy noise, we exclude them when constructing the graph. Table B.13 in the appendix presents a summary of the types and counts of the entities in the MIMIC-III dataset, and the details of each task.

Evaluation Metrics. We evaluate our method with area under the receiver operating curve (AUROC), area under the precision–recall curve (AUPR), accuracy, F1-scores, and Jaccard index, applied as relevant to each task. For robust validation of our results, we employ a five-fold cross-validation strategy in all major experiments. More detailed information on the datasets, tasks and their loss functions, and evaluation metrics is presented in the appendix.

5.2. Compared methods

We compare our method to the following graph data augmentation methods to validate the empirical performance of DemoGraph: LaplacianPE (Dwivedi et al., 2023), RandomWalkPE (Dwivedi et al., 2021), DropEdge (Rong et al., 2020), and DropNode (Feng et al., 2020). For the EHR analysis benchmark, we also include additional competitors as follows: GraphCare (LLM-based) (Jiang et al., 2023), GRU (Medsker & Jain, 2001), Transformer (Vaswani et al., 2017), GRAM (Choi et al., 2017), StageNet (Gao, Xiao, Wang, et al., 2020), Concare (Ma, Zhang, et al., 2020), Adacare (Ma, Gao, et al., 2020), Dr. Agent (Gao, Xiao,

Table 2
Performance [%] of DemoGraph on node classification task for the OGBN-arxiv and OGBN-products datasets.

GNN Archi.	Augmenter	Accuracy	
		OGBN-products	OGBN-arxiv
Graph SAGE	DropNode	54.22 (0.31)	58.42 (0.20)
	DropEdge	55.23 (0.32)	54.83 (0.19)
	RandomWalkPE	OOM	OOM
	LaplacianPE	OOM	OOM
	GraphGPT-std	N/A	62.58
	LLM*	74.40 (0.23)	73.56 (0.06)
	TAPE	81.37 (0.43)	76.72 (0.07)
	GLEM-LM	81.25 (0.15)	74.53 (0.12)
	GLEM-GNN	83.16 (0.19)	75.50 (0.24)
	DemoGraph (Ours)	**84.22 (0.27)**	**76.84 (0.17)**
GAT	DropNode	55.43 (0.34)	57.36 (0.25)
	DropEdge	53.36 (0.37)	58.26 (0.21)
	RandomWalkPE	OOM	OOM
	LaplacianPE	OOM	OOM
	GraphGPT-std	N/A	62.58
	LLM*	74.40 (0.23)	73.56 (0.06)
	TAPE	82.34 (0.36)	**77.50 (0.12)**
	GLEM-LM	OOM	75.45 (0.12)
	GLEM-GNN	OOM	76.97 (0.19)
	DemoGraph (Ours)	**84.00 (0.32)**	**77.18 (0.22)**
GCN	DropNode	56.94 (0.45)	58.57 (0.42)
	DropEdge	54.62 (0.47)	58.15 (0.43)
	RandomWalkPE	OOM	OOM
	LaplacianPE	OOM	OOM
	GraphGPT-std	N/A	62.58
	GraphGPT-stage2	N/A	75.11
	3-HiGCN	N/A	**76.41 (0.53)**
	LLM*	74.40 (0.23)	73.56 (0.06)
	TAPE	79.96 (0.41)	75.20 (0.03)
	GLEM-LM	OOM	75.71 (0.24)
	GLEM-GNN	OOM	75.93 (0.19)
	DemoGraph (Ours)	**82.86 (0.42)**	**76.05 (0.23)**

OOM: out-of-memory. LLM: Using zero-shot ChatGPT with the same prompts of TAPE as the approach, denoted as LLM.

Glass, et al., 2020), and GRASP (Zhang et al., 2021). For drug recommendation, we also include additional competitors: MICRON (Yang, Xiao, Glass, et al., 2021), Safedrug (Yang, Xiao, Ma, Glass, et al., 2021), and MoleRec (Yang, Zeng, Wu, & Yan, 2023). For the large-scale OGBN datasets, additionally, we have included more advanced LLM-based baselines (i.e., GraphGPT (Tang et al., 2023), LLM, TAPE (He et al., 2023), HiGCN (Huang et al., 2024)) and GLEM (Zhao, Qu, et al., 2023). We reimplemented the baseline methods, where details of the implementations and descriptions of the baseline methods can be found in the appendix. For the experiments on generic graph datasets, we do not include all LLM-based methods for comparison, as we find it difficult to implement some of the approaches (e.g. TAPE, HiGCN, GLEM) due to framework adaptation issues. For example, TAPE is restricted to topic-modeling problems as it assumes a topic model, while GLEM focuses on node classification problems and cannot tackle multi-granularity as in the generic datasets.

5.3. Quantitative results

Results on Generic Graph Data. Table 1 presents the node classification results of our proposal compared to existing graph data augmentation methods. Table 2 presents the results on the large-scale OGBN-products and OGBN-arxiv datasets against both traditional and LLM-based competitors. We observe that our method achieves satisfactory performance on generic graph classification datasets, as well as large-scale datasets. Some of the traditional GDA methods that operate on whole graphs failed to generalize to large-scale datasets (i.e., encountered out-of-memory error). Our method obtains a 3% improvement on average over all comparable methods with all four GNN

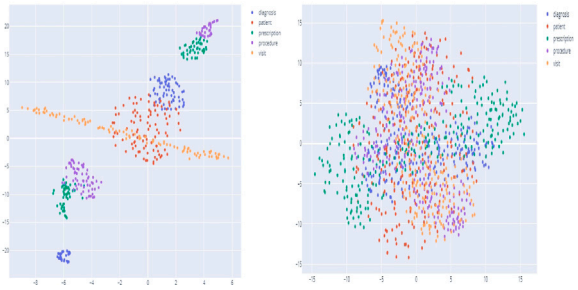

Fig. 4. Visualization of the learned node embeddings w/ (left) and w/o (right) our graph data augmentation, respectively. We use MIMIC-III as the example and color nodes differently by their entity types.

architectures (i.e., GCN (Welling & Kipf, 2016), GAT (Veličković et al., 2017), GIN (Xu et al., 2018), and GraphSAGE (Hamilton et al., 2017a)). This shows evidence that leveraging context knowledge, such as dataset summary and class label information, with LLMs can augment graph data to its true data distribution. We also compare among the comparable methods with different GNN architectures. We observe that our method still performs satisfactorily when different GNN architectures are used, demonstrating the robustness of our method.

Our experiments on the protein-protein interaction dataset demonstrate that DemoGraph enhances the modeling of protein interfaces, which is crucial for accurate protein structure prediction. This can facilitate the identification of potential therapeutic targets and elucidating biological processes.

Results on EHR Data. Table 3 presents the results of different tasks on the MIMIC-III dataset (detailed results with more evaluation metrics are presented in the appendix). We observe that our proposed framework outperforms alternative methods, thereby validating the effectiveness of contextual LLM augmentation and sparsity-aware instruction prompting. In particular, our method outperforms the competitors by 7.4% (in accuracy) in length-of-stay prediction. Our method can even outperform the methods specifically designed for EHR analysis, including GraphCare (Jiang et al., 2023), a similar method using LLM for personalized healthcare. We elaborate the key differences between our method and GraphCare in the appendix. When integrating the enriched context information (e.g., clinical discharge reports, radiology reports, and lab event reports) in real-world EHR datasets, the performance on clinical task prediction can be further improved.

The Effect of Different LLM backbones.

In light of the importance of LLM backbones on the performance of our method, we further study the effects of LLM backbones with differnet capacities. We performed experiments with some renowned black-box LLMs (we access these LLMs only through APIs) shown in Table 4. We observe the differences in model performances, which arise from different training methods and parameter sizes. Nevertheless, our method can maintain satisfactory performance across different LLM backbones, validating its robustness.

5.4. Qualitative results

Embedding Visualization. We visualize the node embeddings of each type of entity to evaluate the performance of feature representation learning. Fig. 4 presents the TSNE plot of the embeddings generated by different methods. The task is readmission prediction on the MIMIC-III dataset with a GAT model. It is observed that the embeddings with DemoGraph are grouped according to their node types, which validates that the embeddings learn the unique representation of each node type, while the embeddings without DemoGraph are noisy and do not present a clear pattern by the node type.

Table 3
Performance of drug recommendation, length of stay, mortality and readmission prediction on MIMIC-III [%]. Standard deviations are shown in brackets.

	Drug Recommendation		Length of Stay		Mortality		Readmission	
Model	AUROC	AUPR	AUROC	Acc.	AUROC	AUPR	AUROC	AUPR
GRU	96.38 (0.1)	64.75 (0.2)	80.32 (0.2)	42.14 (0.6)	61.09 (0.7)	9.65 (1.5)	65.58 (1.1)	68.57 (1.6)
Transformer	95.87 (0.0)	60.19 (0.1)	79.31 (0.8)	41.68 (0.7)	57.20 (1.3)	10.10 (0.9)	63.75 (0.5)	68.92 (0.1)
Deepr	96.09 (0.0)	62.48 (0.1)	78.02 (0.4)	39.31 (1.2)	60.80 (0.4)	13.20 (1.1)	66.50 (0.4)	68.80 (0.9)
GRAM	94.20 (0.0)	76.70 (0.1)	78.02 (0.4)	39.31 (1.2)	60.40 (0.9)	11.40 (0.7)	64.30 (0.4)	67.20 (0.8)
Concare	95.78 (0.1)	61.67 (0.3)	80.27 (0.3)	42.04 (0.6)	61.98 (1.8)	9.67 (1.5)	65.28 (1.1)	66.67 (1.9)
Dr. Agent	96.41 (0.1)	64.16 (0.5)	79.45 (0.6)	41.40 (0.5)	57.52 (0.4)	9.66 (0.8)	64.86 (2.6)	67.41 (1.0)
AdaCare	95.86 (0.0)	60.76 (0.0)	78.73 (0.4)	40.70 (0.8)	58.40 (1.4)	11.10 (0.4)	65.70 (0.3)	68.60 (0.6)
StageNet	96.05 (0.0)	62.43 (2.4)	77.94 (0.2)	40.70 (0.8)	61.50 (0.7)	12.40 (0.3)	66.70 (0.4)	69.30 (0.6)
GRASP	96.01 (0.1)	62.53 (0.3)	78.97 (0.4)	40.66 (0.3)	59.20 (1.4)	9.90 (1.1)	66.30 (0.6)	69.20 (0.4)
DropNode	97.60 (0.2)	81.41 (0.1)	81.10 (0.5)	41.81 (1.1)	58.06 (0.9)	9.46 (1.7)	64.48 (0.8)	67.75 (0.4)
DropEdge	95.61 (0.1)	72.32 (0.3)	78.41 (0.3)	39.98 (0.8)	57.85 (0.8)	10.34 (1.5)	62.11 (0.6)	67.46 (0.5)
RandomWalkPE	94.89 (0.1)	63.86 (0.2)	78.01 (0.4)	39.47 (0.9)	57.15 (1.2)	9.76 (0.9)	66.20 (0.7)	59.58 (0.6)
LaplacianPE	95.26 (0.2)	69.34 (0.3)	78.22 (0.3)	40.02 (0.9)	57.65 (1.1)	10.05 (1.2)	65.71 (0.6)	63.43 (0.8)
GraphCare	95.00 (0.0)	78.50 (0.2)	79.40 (0.3)	41.90 (0.2)	66.60 (1.1)	14.30 (0.8)	68.10 (0.6)	71.50 (0.7)
DemoGraph (Ours)	**98.54 (0.2)**	**83.89 (0.1)**	**82.68 (0.2)**	**45.28 (1.0)**	**67.79 (0.6)**	**16.09 (1.6)**	**68.97 (0.4)**	**73.92 (0.4)**

Table 4
Performance of mortality and readmission prediction on MIMIC-III [%] with different LLM backbones. Standard deviations are shown in brackets.

	Mortality		Readmission	
Models	AUROC	AUPR	AUROC	AUPR
GraphCare (GPT-4, KG method)	66.6 (1.1)	14.3 (0.8)	68.1 (0.6)	71.5 (0.7)
DemoGraph (LLaMA-3.1-8B)	66.5 (0.9)	14.7 (1.1)	68.0 (0.7)	71.1 (0.8)
DemoGraph (Claude-3-Opus)	66.9 (1.0)	15.7 (0.8)	69.0 (0.6)	73.0 (0.7)
DemoGraph (LLaMA-3.1-70B)	67.1 (0.8)	15.9 (1.0)	69.2 (0.5)	73.1 (0.6)
DemoGraph (GPT-4, **original**)	67.7 (0.6)	16.0 (1.6)	69.0 (0.4)	73.9 (0.4)
DemoGraph (LLaMA-3.1-405B)	67.9 (0.7)	16.3 (1.0)	69.4 (0.5)	73.8 (0.6)
DemoGraph (Claude-3.5-Sonnet)	68.0 (0.5)	16.3 (0.8)	69.6 (0.3)	73.9 (0.5)
DemoGraph (GPT-4o-mini)	68.0 (0.6)	16.4 (0.7)	69.5 (0.4)	74.0 (0.4)
DemoGraph (GPT-4o)	68.1 (0.5)	16.3 (0.8)	69.6 (0.3)	**74.1 (0.5)**
DemoGraph (DeepSeek-V3)	**68.6 (0.6)**	**17.0 (0.9)**	**69.8 (0.6)**	**74.1 (0.4)**

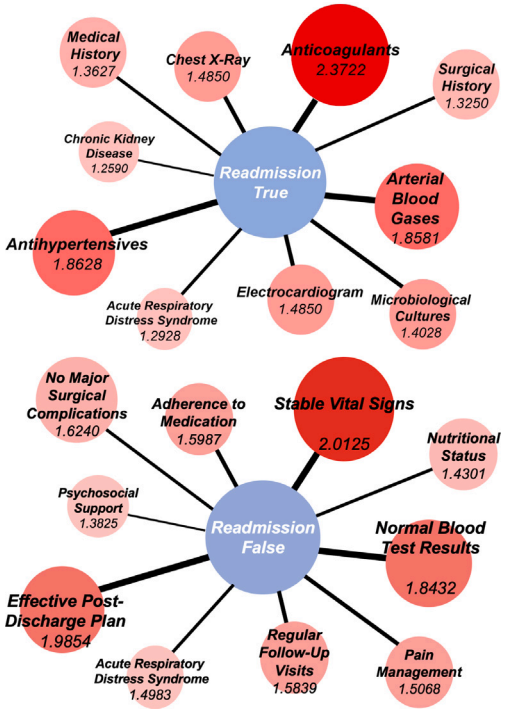

Fig. 5. Interpretability visualization of DemoGraph: (Top) Readmission node predicted as True, (Down) Case where readmission is False: a visit node (blue) and related concept nodes (red), with attention scores, visualized in size/shade of red nodes.

Table 5
Performance w/ and w/o augmentation from KG, and w/ a biased KG from another dataset (i.e. PPI), repectively.

	w/o KG		w/ KG		w/ PPI KG	
Dataset	Acc.	F1	Acc.	F1	Acc.	F1
Cora	82.10	81.66	**83.60**	**83.64**	73.70	73.83
Actor	30.33	27.90	**32.21**	**28.91**	30.19	27.82
Citeseer	72.10	69.60	**73.10**	**72.46**	63.40	64.68

Network Interpretation. The incorporation of contextual learning enhances the capability of the model by enabling a nuanced understanding and interpretation of the graph data at a deeper level. We analyze the interpretability of our model by considering a specific visit node in the MIMIC-III dataset. As shown in Fig. 5, the following are the top augmented corrections (i.e., with the highest attention scores) that exemplify the importance of specific clinical concepts influencing the "readmission is true" prediction: Antihypertensives (2.3722), Anticoagulants (1.8628), and arterial blood gases (1.8581), where the computed attention scores are shown below the node name. It is observed that the augmentation process can impute context-related concepts so that GAT can select the most important ones. This provides interpretations for the predictive process. This is especially beneficial in the clinical decision context since the enriched open-world knowledge can inspire clinicians with the embedded concepts, and enhance the understanding of patients' behaviors and the potential reasons for certain diseases.

5.5. Ablated analysis

The Effect of Augmented KGs. We study the effect of augmented KGs on downstream task performance (Table 5), including three scenarios:

Table 6
Performance of node classification (using GAT) with and without dynamic merging, respectively.

	Cora		Actor		PPI		Citeseer	
Merging	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Static	83.30	83.43	31.45	28.01	96.82	94.67	72.20	71.73
Dynamic	**83.60**	**83.64**	**32.21**	**28.91**	**98.28**	**97.20**	**73.10**	**72.46**

Table 7
Performance of node classification (using GAT) with different numbers of edges per concept generated by the KG.

$\lvert\mathcal{E}^{\text{conn}}\rvert$	Cora		Actor		PPI		Citeseer	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
0	81.3	80.7	30.3	28.0	91.6	97.1	72.1	69.6
3	83.6	83.6	32.2	28.9	96.4	97.2	73.1	72.5
30	79.3	79.4	31.0	28.8	97.5	97.2	68.5	68.3
100	75.4	75.5	30.9	28.4	98.3	97.2	66.2	66.2

Table 8
Performance of our framework on Cora node classification with different granularity levels s, and with or without IFT, respectively. We denote s_1 as the class type level, s_0 as the dataset level, and $s_1 + s_0$ as a multi-granularity scheme merging these two levels.

IFT	$s = s_1$		$s = s_0$		$s = s_0 + s_1$	
	Acc.	F1	Acc.	F1	Acc.	F1
w/o IFT	81.40	81.53	82.17	82.05	81.00	81.07
w/ IFT	83.20	83.26	**83.60**	**83.64**	83.15	83.25

Table 9
Performance in terms of accuracy (%) of our framework on node classification with different numbers of layers L, using GCN and GAT.

	GCN			GAT		
L	Cora	Actor	Citeseer	Cora	Actor	Citeseer
1	81.40	31.91	71.77	83.30	32.21	72.70
2	81.50	32.41	73.10	83.60	29.21	73.10
3	82.90	31.45	70.45	82.10	28.49	72.10
4	80.50	30.54	70.04	81.70	28.20	71.70

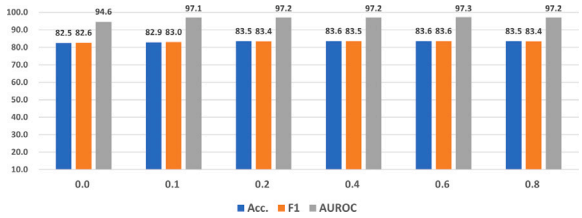

Fig. 6. Performance of our method on Cora node classifications with respect to different dropout ratios, with GAT as the GNN architecture.

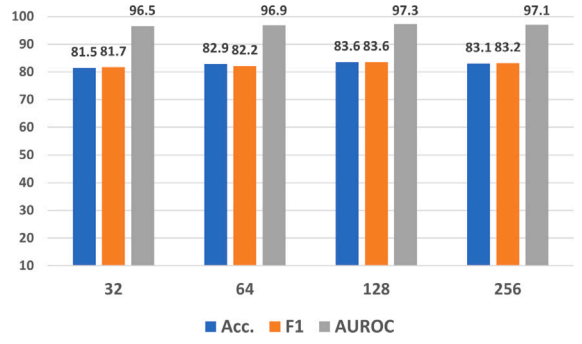

Fig. 7. Performance of our method on Cora node classifications with respect to different numbers of hidden dimensions, with GAT as the GNN architecture.

with KG, without KG, and with a biased (or wrong) KG augmented from another dataset (i.e. PPI). It is observed that the model performs worse than the baseline (i.e., w/o any augmentations) when the wrong context is applied, indicating a biased augmented graph. On the other hand, improved performance is observed when a context-driven KG is applied, thus validating the effectiveness of our method. A visualization of the effect of DemoGraph on node embeddings can also be found in Fig. 4.

The Effect of Dynamic Merging. We evaluate the contribution of the dynamic merging schema, as summarized in Table 6, where static merging means that the KG are merged into $\mathcal{G}_0$ offline before training. We observe that the performance improved on all generic graph datasets with dynamic merging, which validates the contributions of the schema.

The Effect of Sparsity Control. We demonstrate how different levels of sparsity affect the performance of graph data augmentation. We control the level of sparsity using the number of edges per concept $|\mathcal{E}^{\text{conn}}|$ used for KG generation. Table 7 presents the results of this study. Given a fixed number of concepts, the performance improves when $|\mathcal{E}^{\text{conn}}|$ increases, demonstrating the effectiveness of graph merging. However, when $|\mathcal{E}^{\text{conn}}|$ is too large compared to the original graph size, the augmented graph would be biased from too many noisy connections, and hence the observed performance deteriorates.

The Influence of Different Granularity and Instruction Fine-tuning. We evaluate the influence of different granularity and instruction fine-tuning (IFT) on augmentation performance. From Table 8, it is observed that the performance is improved when an appropriate s is chosen, while adopting a multi-granularity $(s_0 + s_1)$ could potentially lead to over-sparsification. With KG concepts pruned by IFT, the performance is consistently improved on different granularity levels.

The Influence of Number of GNN Layers. We evaluate the performance of our method with different numbers of GNN layers, as summarized in Table 9. We observe that in general a better performance is obtained when the number of layers is small. The performance slightly deteriorates as the number of layers increases more than two layers, indicating the potential over-smoothing problem. Other experiments on relatively fine-grained hyperparameters, such as the dropout rate, number of hidden dimensions, and number of attention heads for GAT, are presented in the appendix.

Dropout Ratios. Since graph learning is difficult to optimize and easy to lead to overfitting, we adopt dropout as the default regularizer for all benchmark methods. We further study the effects of different dropout rates, Fig. 6 presents the results. We observe that our method is in general robust to changes in dropout rates while being optimized when the dropout rate is 0.6. However, a large dropout rate would lead to over-sparsification of neural network weights and important features being dropped, hindering the predictive performance.

Number of hidden dimensions. We benchmark our method with respect to different hidden dimensions. Fig. 7 presents the results of this study.

We observe our method is overall robust to different numbers of hidden dimensions. In general, a larger number of hidden dimensions leads to better classification performance.

Number of attention heads. We benchmark the performance of our method with respect to different attention heads, as summarized in Fig. 8. We observe that the performance is overall improving with the

Table 10
Analysis of time complexity of training time on the `ogbn-arxiv` dataset.

Method	DemoGraph (Ours) on GAT	TAPE	GraphGPT-stage-2	GraphGPT-stage-1
Training Time (min)	89	192	224	1325

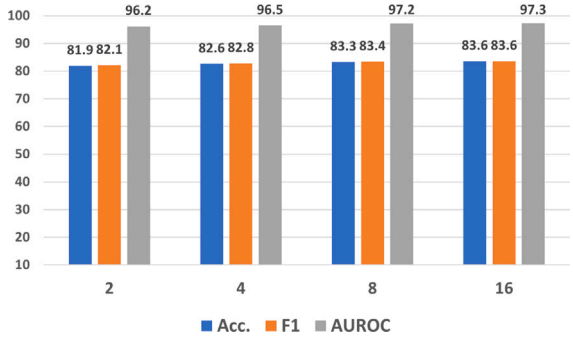

Fig. 8. Performance of our method on Cora node classifications with respect to different heads of GAT.

number of heads increases, while a larger number of heads (e.g., 32) would lead to a heavier memory burden under the current hardware settings.

6. Discussion

Fairness and Privacy Discussion

In the context of EHR research, fairness and privacy are critical concerns that warrant thorough discussion, particularly when incorporating LLMs for data augmentation. Although the use of LLMs can enhance the capture of nuanced clinical relationships, there is a risk that inherent biases in the training data may be amplified, thereby exacerbating disparities across ethnicities and genders. Additionally, the integration of sensitive patient information within these models introduces potential privacy risks, emphasizing the need for robust anonymization and data protection protocols. Prior studies have provided valuable insights into these challenges, with empirical investigations characterizing fairness in clinical risk prediction (Liu, Li, et al., 2022) and proposing deconfounding approaches to mitigate health disparities in EHRs (Liu, Li, & Yu, 2023; Pfohl, Foryciarz, & Shah, 2021). Moreover, the nature of adaptability under the black-box settings of DemoGraph inherently protects the privacy of patients, as the framework leverages latent KGs extracted via external LLM interfaces without direct access to raw patient data, thereby ensuring that sensitive information remains anonymized. Additionally, in line with the insights provided by Pfohl et al. (2024) in their empirical characterization of fair machine learning for clinical risk prediction, we recognize that ensuring equitable predictions requires careful evaluation of how biases may affect disparate patient groups. Addressing these concerns within the current framework would not only strengthen its ethical foundation but also enhance its applicability in real-world clinical settings.

Time Complexity Analysis. Since we generate the KGs offline using the OpenAI API of gpt-4-0125-preview (OpenAI, 2023) (our method works under a black-box setting), this process only needs to be performed once for each dataset. The additional complexity arises from the dynamic merging process, which needs to be repeated at each optimization step. However, the time complexity of this step is trivial compared to the forward passing of GNNs. Therefore, it only increases the overall time complexity on a minor level.

Table 10 below shows the quantitative analysis of the training time complexity on the `ogbn-arxiv` dataset.

Efficiency through Single Query and Reuse. Our prompting paradigm avoids manual prompt customization for adaptations to different datasets, thereby reducing human labor costs. Our method necessitates only a single query to the LLM, with KGs and significant concept nodes stored for subsequent reuse. Our query process can be efficiently completed in 37.6 s in average for the large-scale ogbn-arxiv dataset. This approach not only enhances efficiency but also reduces the number of API calls, thereby saving the cost of commercial LLMs. Additionally, we have provided the responses from the LLMs gained in our experiments for the public use.

7. Conclusion

We propose a novel framework for graph data augmentation, namely DemoGraph, which leverages the open-world knowledge in LLMs to perform context-driven graph data augmentation. Our method directly operates on knowledge graphs constructed from LLM outputs and does not require access to model weights and features, which enables democratization to most of the closed-access LLMs. To tackle the sparsity induced by generated knowledge graphs, we design a granularity-aware prompting strategy to control the sparsity while maximizing the utility of domain knowledge. Experiments on generic graph datasets and a medical records dataset with an array of GNN architectures validate that our method can better augment the graph data than existing methods. Ablation analysis on key components and hyperparameters of our method validates the significance of our method and robustness to variations. Our method also has a wide range of potential application fields beyond medical record analysis such as molecular chemistry, recommendation, computational biology, social networks, and citation networks etc.

CRediT authorship contribution statement

Yushi Feng: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Tsai Hor Chan:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Guosheng Yin:** Writing – review & editing. **Lequan Yu:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author(s) used ChatGPT in order to polish the language. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the Research Grants Council of Hong Kong (27206123 and T45-401/22-N), in part by the Hong Kong Innovation and Technology Fund (ITS/273/22 and ITS/274/22), in part by the National Natural Science Foundation of China (No. 62201483), and in part by Guangdong Natural Science Fund (No. 2024A1515011875).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.neunet.2025.107777.

Data availability

The codes for reproducing this work are available at https://github.com/HKU-MedAI/DemoGraph. The dataset MIMIC-III is available at https://physionet.org/content/mimiciii/1.4/ and MIMIC-IV is available at https://physionet.org/content/?topic=mimic-iv.

References

Cai, X., Huang, C., Xia, L., & Ren, X. (2023). LightGCL: Simple yet effective graph contrastive learning for recommendation. In *The eleventh international conference on learning representations*.

Chan, T. H., Wong, C. H., Shen, J., & Yin, G. (2023). Source-aware embedding training on heterogeneous information networks. *Data Intelligence*, 1–14.

Chen, Z., Feng, Y., He, C., Deng, Y., Pu, H., & Li, B. (2025). IPAD: inverse prompt for AI detection – A robust and explainable LLM-generated text detector. arXiv preprint arXiv:2502.15902.

Choi, E., Bahadori, M. T., Song, L., Stewart, W. F., & Sun, J. (2017). GRAM: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 787–795).

Choi, E., Xiao, C., Stewart, W., & Sun, J. (2018). Mime: Multilevel medical embedding of electronic health records for predictive healthcare. *Advances in Neural Information Processing Systems*, *31*.

Del Fiol, G., Curtis, C., Cimino, J. J., Iskander, A., Kalluri, A. S., Jing, X., et al. (2013). Disseminating context-specific access to online knowledge resources within electronic health record systems. *Studies in Health Technology and Informatics*, *192*, 672.

Ding, K., Xu, Z., Tong, H., & Liu, H. (2022). Data augmentation for deep graph learning: A survey. *ACM SIGKDD Explorations Newsletter*, *24*(2), 61–77.

Dwivedi, V. P., Joshi, C. K., Luu, A. T., Laurent, T., Bengio, Y., & Bresson, X. (2023). Benchmarking graph neural networks. *Journal of Machine Learning Research*, *24*(43), 1–48.

Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y., & Bresson, X. (2021). Graph neural networks with learnable structural and positional representations. In *International conference on learning representations*.

Evans, R. S. (2016). Electronic health records: then, now, and in the future. *Yearbook of Medical Informatics*, *25*(S 01), S48–S61.

Feng, W., Zhang, J., Dong, Y., Han, Y., Luan, H., Xu, Q., et al. (2020). Graph random neural networks for semi-supervised learning on graphs. *Advances in Neural Information Processing Systems*, *33*, 22092–22103.

Franceschi, L., Niepert, M., Pontil, M., & He, X. (2019). Learning discrete structures for graph neural networks. In *International conference on machine learning* (pp. 1972–1982). PMLR.

Gao, J., Xiao, C., Glass, L. M., & Sun, J. (2020). Dr. Agent: Clinical predictive model via mimicked second opinions. *Journal of the American Medical Informatics Association*, *27*(7), 1084–1091.

Gao, J., Xiao, C., Wang, Y., Tang, W., Glass, L. M., & Sun, J. (2020). Stagenet: Stage-aware neural networks for health risk prediction. In *Proceedings of the web conference 2020* (pp. 530–540).

Hamilton, W., Ying, Z., & Leskovec, J. (2017a). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, *30*.

Hamilton, W., Ying, Z., & Leskovec, J. (2017b). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, *30*.

He, X., Bresson, X., Laurent, T., Perold, A., LeCun, Y., & Hooi, B. (2023). Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *The twelfth international conference on learning representations*.

Hsu, W., Taira, R. K., El-Saden, S., Kangarloo, H., & Bui, A. A. (2012). Context-based electronic health record: toward patient specific healthcare. *IEEE Transactions on Information Technology in Biomedicine*, *16*(2), 228–234.

Hu, Z., Dong, Y., Wang, K., & Sun, Y. (2020). Heterogeneous graph transformer. In *Proceedings of the web conference 2020* (pp. 2704–2710).

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., et al. (2020). Open graph benchmark: Datasets for machine learning on graphs. arXiv preprint arXiv:2005.00687.

Huang, T., Xu, K., & Wang, D. (2020). DA-HGT: Domain adaptive heterogeneous graph transformer. arXiv preprint arXiv:2012.05688.

Huang, Y., Zeng, Y., Wu, Q., & Lü, L. (2024). Higher-order graph convolutional network with flower-petals laplacians on simplicial complexes. *vol. 38*, In *Proceedings of the AAAI conference on artificial intelligence* (11), (pp. 12653–12661).

Jiang, P., Xiao, C., Cross, A., & Sun, J. (2023). GraphCare: Enhancing healthcare predictions with open-world personalized knowledge graphs. arXiv preprint arXiv:2305.12788.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., et al. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*, *3*(1), 1–9.

Kojima, R., Ishida, S., Ohta, M., Iwata, H., Honma, T., & Okuno, Y. (2020). kGCN: a graph-based deep learning framework for chemical structures. *Journal of Cheminformatics*, *12*, 1–10.

Li, Q., Han, Z., & Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. *vol. 32*, In *Proceedings of the AAAI conference on artificial intelligence*. (1).

Liu, Z., Li, X., Peng, H., He, L., & Philip, S. Y. (2020). Heterogeneous similarity graph neural network on electronic health records. In *2020 IEEE international conference on big data (big data)* (pp. 1196–1205). IEEE.

Liu, Z., Li, X., & Yu, P. S. (2022). Mitigating health disparities in EHR via deconfounder. In *BCB '22: 13th ACM international conference on bioinformatics, computational biology and health informatics, northbrook, illinois, USA, August 7 - 10, 2022* (pp. 6:1–6:6). ACM, http://dx.doi.org/10.1145/3535508.3545516.

Liu, Z., Li, X., & Yu, P. S. (2023). A counterfactual fair model for longitudinal electronic health records via deconfounder. In G. Chen, L. Khan, X. Gao, M. Qiu, W. Pedrycz, & X. Wu (Eds.), *IEEE international conference on data mining, ICDM 2023, shanghai, China, December 1-4, 2023* (pp. 1175–1180). IEEE, http://dx.doi.org/10.1109/ICDM58522.2023.00144.

Liu, S., Ying, R., Dong, H., Li, L., Xu, T., Rong, Y., et al. (2022). Local augmentation for graph neural networks. In *International conference on machine learning* (pp. 14054–14072). PMLR.

Ma, F., Chitta, R., Zhou, J., You, Q., Sun, T., & Gao, J. (2017). Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1903–1911).

Ma, L., Gao, J., Wang, Y., Zhang, C., Wang, J., Ruan, W., et al. (2020). Adacare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration. *vol. 34*, In *Proceedings of the AAAI conference on artificial intelligence* (01), (pp. 825–832).

Ma, F., You, Q., Xiao, H., Chitta, R., Zhou, J., & Gao, J. (2018). Kame: Knowledge-based attention model for diagnosis prediction in healthcare. In *Proceedings of the 27th ACM international conference on information and knowledge management* (pp. 743–752).

Ma, L., Zhang, C., Wang, Y., Ruan, W., Wang, J., Tang, W., et al. (2020). Concare: Personalized clinical feature embedding via capturing the healthcare context. *vol. 34*, In *Proceedings of the AAAI conference on artificial intelligence* (01), (pp. 833–840).

Medsker, L. R., & Jain, L. (2001). Recurrent neural networks. *Design and Applications*, *5*, 64–67.

OpenAI (2023). GPT-4 technical report. arXiv:2303.08774.

Park, H., Lee, S., Kim, S., Park, J., Jeong, J., Kim, K.-M., et al. (2021). Metropolis-hastings data augmentation for graph neural networks. *Advances in Neural Information Processing Systems*, *34*, 19010–19020.

Park, J., Shim, H., & Yang, E. (2022). Graph transplant: Node saliency-guided graph mixup with local structure preservation. *vol. 36*, In *Proceedings of the AAAI conference on artificial intelligence* (7), (pp. 7966–7974).

Pfohl, S. R., Cole-Lewis, H., Sayres, R., Neal, D., Asiedu, M., Dieng, A., et al. (2024). A toolbox for surfacing health equity harms and biases in large language models. *Nature Medicine*, *30*(12), 3590–3600.

Pfohl, S. R., Foryciarz, A., & Shah, N. H. (2021). An empirical characterization of fair machine learning for clinical risk prediction. *J. Biomed. Informatics*, *113*, Article 103621. http://dx.doi.org/10.1016/J.JBI.2020.103621.

Qiu, J., Chen, Q., Dong, Y., Zhang, J., Yang, H., Ding, M., et al. (2020). Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1150–1160).

Rong, Y., Huang, W., Xu, T., & Huang, J. (2019). Dropedge: Towards deep graph convolutional networks on node classification. arXiv preprint arXiv:1907.10903.

Rong, Y., Huang, W., Xu, T., & Huang, J. (2020). DropEdge: Towards deep graph convolutional networks on node classification. arXiv:1907.10903.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Berg, R. v. d., Titov, I., & Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European semantic web conference* (pp. 593–607). Springer.

Shi, C., Hu, B., Zhao, W. X., & Philip, S. Y. (2018). Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *31*(2), 357–370.

Simonovsky, M., & Komodakis, N. (2018). Graphvae: Towards generation of small graphs using variational autoencoders. In *Artificial neural networks and machine learning–ICANN 2018: 27th international conference on artificial neural networks, rhodes, Greece, October 4-7, 2018, proceedings, part i 27* (pp. 412–422). Springer.

Suresh, S., Li, P., Hao, C., & Neville, J. (2021). Adversarial graph augmentation to improve graph contrastive learning. *Advances in Neural Information Processing Systems*, *34*, 15920–15933.

Tang, J., Yang, Y., Wei, W., Shi, L., Su, L., Cheng, S., et al. (2023). GraphGPT: Graph instruction tuning for large language models. arXiv:2310.13023.

Tang, J., Yang, Y., Wei, W., Shi, L., Su, L., Cheng, S., et al. (2024). GraphGPT: Graph instruction tuning for large language models. arXiv:2310.13023.

Topping, J., Di Giovanni, F., Chamberlain, B. P., Dong, X., & Bronstein, M. M. (2021). Understanding over-squashing and bottlenecks on graphs via curvature. arXiv preprint arXiv:2111.14522.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.

Veličković, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., & Hjelm, R. D. (2018). Deep graph infomax. arXiv preprint arXiv:1809.10341.

Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., et al. (2019). Heterogeneous graph attention network. In *The world wide web conference* (pp. 2022–2032).

Wang, Y., Wang, W., Liang, Y., Cai, Y., Liu, J., & Hooi, B. (2020). Nodeaug: Semi-supervised node classification with data augmentation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 207–217).

Wei, W., Ren, X., Tang, J., Wang, Q., Su, L., Cheng, S., et al. (2024). Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM international conference on web search and data mining* (pp. 806–815).

Welling, M., & Kipf, T. N. (2016). Semi-supervised classification with graph convolutional networks. In *J. international conference on learning representations (ICLR 2017)*.

West, P., Bhagavatula, C., Hessel, J., Hwang, J. D., Jiang, L., Bras, R. L., et al. (2021). Symbolic knowledge distillation: from general language models to commonsense models. arXiv preprint arXiv:2110.07178.

Wu, T., Ren, H., Li, P., & Leskovec, J. (2020). Graph information bottleneck. *Advances in Neural Information Processing Systems*, *33*, 20437–20448.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks? In *International conference on learning representations*.

Yang, L., Kang, Z., Cao, X., Jin, D., Yang, B., & Guo, Y. (2019). Topology optimization based graph convolutional network.. In *IJCAI* (pp. 4054–4061).

Yang, S., Song, G., Jin, Y., & Du, L. (2020). Domain adaptive classification on heterogeneous information networks. In *IJCAI* (pp. 1410–1416).

Yang, C., Xiao, C., Glass, L., & Sun, J. (2021). Change matters: Medication change prediction with recurrent residual networks. In *30th international joint conference on artificial intelligence, IJCAI 2021* (pp. 3728–3734). International Joint Conferences on Artificial Intelligence.

Yang, C., Xiao, C., Ma, F., Glass, L., & Sun, J. (2021). Safedrug: Dual molecular graph encoders for recommending effective and safe drug combinations. arXiv preprint arXiv:2105.02711.

Yang, N., Zeng, K., Wu, Q., & Yan, J. (2023). MoleRec: Combinatorial drug recommendation with substructure-aware molecular representation learning. In *Proceedings of the ACM web conference 2023* (pp. 4075–4085).

You, Y., Chen, T., Wang, Z., & Shen, Y. (2020). When does self-supervision help graph convolutional networks? In *International conference on machine learning* (pp. 10871–10880). PMLR.

Yun, S., Jeong, M., Kim, R., Kang, J., & Kim, H. J. (2019). Graph transformer networks. *Advances in Neural Information Processing Systems*, *32*.

Zhang, X., Bosselut, A., Yasunaga, M., Ren, H., Liang, P., Manning, C. D., et al. (2022). Greaselm: Graph reasoning enhanced language models for question answering. arXiv preprint arXiv:2201.08860.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). Mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412.

Zhang, C., Gao, X., Ma, L., Wang, Y., Wang, J., & Tang, W. (2021). GRASP: generic framework for health status representation learning based on incorporating knowledge from similar patients. *vol. 35*, In *Proceedings of the AAAI conference on artificial intelligence* (1), (pp. 715–723).

Zhao, J., Qu, M., Li, C., Yan, H., Liu, Q., Li, R., et al. (2023). Learning on large-scale text-attributed graphs via variational inference. In *The eleventh international conference on learning representations, ICLR 2023, kigali, rwanda, May 1-5, 2023*. OpenReview.net, URL https://openreview.net/forum?id=q0nmYciuuZN.

Zheng, C., Zong, B., Cheng, W., Song, D., Ni, J., Yu, W., et al. (2020). Robust graph representation learning via neural sparsification. In *International conference on machine learning* (pp. 11458–11468). PMLR.