# GT2Vec: Large Language Models for Knowledge Graph Augmented Text Embedding

Anonymous Author(s)


## Abstract

Graph-structured information offers rich contextual information that can enhance language models by providing structured relationships and hierarchies, leading to more expressive embeddings for various applications such as retrieval, question answering, and classification. However, existing methods for integrating graph and text embeddings, often based on Multi-layer Perceptrons (MLPs) or shallow transformers, are limited in their ability to fully exploit the heterogeneous nature of these modalities. To overcome this, we propose GT2Vec, a simple yet effective framework that leverages Large Language Models (LLMs) to jointly encode text and graph data. Specifically, GT2Vec employs an MLP adapter to project graph embeddings into the same space as text embeddings, allowing the LLM to process both modalities jointly. Unlike prior work, we also introduce contrastive learning to align the graph and text spaces more effectively, thereby improving the quality of learned joint embeddings. Empirical results across six datasets spanning three tasks—knowledge graph-contextualized question answering, graph-text pair classification, and retrieval—demonstrate that GT2Vec consistently outperforms existing baselines, achieving significant improvements across multiple datasets. These results highlight GT2Vec's effectiveness in integrating graph and text data. Ablation studies further validate the effectiveness of our method.



## Keywords

Large Language Models, Representation Learning, Graph Neural Networks, Contrastive Learning

**ACM Reference Format:**
Anonymous Author(s). 2018. GT2Vec: Large Language Models for Knowledge Graph Augmented Text Embedding. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX


## 1 Introduction

In the realm of natural language processing (NLP), text embeddings play a pivotal role by transforming textual information into numerical representations, which facilitate a multitude of machine learning applications on the Web, such as question answering (QA) [7, 49], retrieval [4, 39, 44, 59], and classification tasks [7, 59]. These applications can benefit from the integration of graph-structured data to enhance the capabilities of NLP systems, by providing contextual information or by augmenting the original tasks with additional information. For example, prior work has shown that a QA system that includes a knowledge graph as input can leverage the relationships and hierarchies within the graph to more accurately understand and handle complex queries [64, 68]. To effectively integrate these two modalities, it is essential to develop methods to learn joint embeddings of graph-structured data and text data. Such embeddings can provide a unified representation that captures important information from both modalities, leading to performance improvements across various NLP tasks.

Prior research has introduced various ways to learn joint embeddings of text and graph-structured data for embedding tasks [11, 32, 64, 68]. These methods typically utilize either a Multi-layer Perceptron (MLP) or a shallow transformer [56] to integrate text features and graph embeddings encoded respectively by language models (LMs) [7, 38] and graph neural networks (GNNs) [28, 57, 63]. Despite their effectiveness, these approaches demonstrate a restricted capacity to fuse the features of the two modalities. The primary limitation arises from the limited ability of MLPs and shallow transformers to manage the high-dimensional and heterogeneous nature of joint embeddings, which can result in sub-optimal utilization of the rich contextual information from text and graph data. Recently, large language models (LLMs) have demonstrated significant potential for integrating and understanding modalities beyond just text. A representative example of this capability is in Vision-Language Models (VLMs) [2, 31, 37, 71], where visual tokens are combined with textual input and processed together by LLMs. This integration leverages the powerful capabilities of LLMs to handle multimodal data, allowing for a more holistic understanding of content that spans different forms of information. Inspired by this, we explore the potential of employing LLMs to better integrate text and graph-structured data, aiming to overcome the limitations observed from current approaches.

In this paper, we present GT2Vec, a simple yet effective framework to learn joint embeddings of text and graph data, leveraging the advanced capabilities of LLMs to address the limitations of previous approaches. Our method seamlessly integrates graph and text embeddings within the LLM framework, enhancing their alignment and interaction. Specifically, we transform graph embeddings into the same space as text embeddings using a multi-layer perceptron (MLP) adapter, enabling the LLM to process both modalities together. Additionally, we propose a contrastive learning strategy to better align the graph and text spaces, ensuring that the model learns richer representations of the combined data. Our extensive empirical analysis across a variety of NLP tasks highlights the key advantage of GT2Vec: the ability to leverage LLMs' strong language understanding and reasoning capabilities to process multimodal data, thus providing a more holistic and nuanced representation of both graph-structured and textual information. This integration




*Conference acronym 'XX, June 03–05, 2018, Woodstock, NY*





ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

leads to significant improvements in various NLP tasks, demonstrating the potential of GT2Vec to push the boundaries of joint text and graph-embedding techniques.

Our contributions are summarized as follows:

- **Integration of LLMs for Joint Embeddings**: We propose GT2Vec framework that leverages the strengths of LLMs to align and integrate graph and text embeddings. GT2Vec effectively captures the rich contextual information of both modalities, enabling more robust joint representations.
- **Contrastive Learning for Graph-Text Alignment**: We introduce a contrastive learning mechanism to explicitly align graph and text embeddings, enabling the model to better integrate the two modalities.
- **Extensive Empirical Validation**: We conduct extensive experiments on six datasets spanning three different tasks: knowledge graph (KG)-contextualized QA, graph-text pair classification, and retrieval tasks. GT2Vec achieves superior performance on all the three tasks, demonstrating its ability to effectively integrate graph and text data for enhanced multi-modal representation learning.

## 2 Problem Statement

Given an input text $x$ and its corresponding graph context $\mathcal{G}$, where $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ consists of a set of nodes $\mathcal{V}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ that connect nodes via relationships $\mathcal{R}$, our goal is to extract joint embeddings $\phi(x, \mathcal{G})$. Here, $\phi$ is a learned function that maps the input text $x$ and graph $\mathcal{G}$ into a unified vector representation, capturing the multimodal information. These joint embeddings $\phi(x, \mathcal{G})$ are then used for downstream tasks. In this paper, we focus on three specific downstream tasks: multi-choice QA contextualized by KG, graph-text pair classification, and retrieval tasks.

### 2.1 KG-Contextualized QA

Given a question $q$ in text form, KG context $\mathcal{G}$, and answer candidate set with $n$ choices $\boldsymbol{a} = \{a_1, \cdots, a_n\}$, KG-contextualized QA tasks aim to find the correct textual answer $a_i$ from $\boldsymbol{a}$. Each choice $a_i$ is first concatenated with the question $q$, leading to the input $x = [q, a_i]$, where $[\cdot, \cdot]$ denotes the concatenation of text. We then extract the joint embeddings $\phi(x, \mathcal{G})$ which is fed into a MLP layer to calculate scores. The choice with the highest score is selected as the prediction.

### 2.2 Graph-Text Pair Classification

In the task of graph-text pair classification, the objective is to determine the relevance between a given graph, represented as $\mathcal{G}$, and a corresponding textual description $x$. This involves assessing whether the content and structure of $\mathcal{G}$ are accurately reflected or described by $x$.

To achieve this, we first compute the joint embeddings $\phi(x, \mathcal{G})$ which capture the features and relationships contained in both the text and the graph. Once the joint embeddings are obtained, they are input into a MLP classifier. It outputs a prediction score which measures the likelihood of the graph $\mathcal{G}$ matching the text $x$.

The significance of this task lies in its ability to improve the quality of training data for tasks such as generating text from knowledge graphs and vice versa [6, 27, 30]. By accurately classifying graph-text pairs, the model can help reduce noise in training data, which in turn improves the overall performance of generation tasks [30].

### 2.3 Retrieval

For the retrieval task, given a textual query $q$ accompanied by its graph context $\mathcal{G}_q$, the goal is to retrieve the most relevant candidate from a set of text-based options. Each candidate $c_i$ in the candidate set $\boldsymbol{c} = \{c_1, \cdots, c_m\}$ also has an associated graph context $\mathcal{G}_{c_i}$. The task involves comparing the query-graph pair $(q, \mathcal{G}_q)$ against each candidate-graph pair $(c_i, \mathcal{G}_{c_i})$.

To achieve this, we first generate the joint embeddings $\phi(q, \mathcal{G}_q)$ for the query and $\phi(c_i, \mathcal{G}_{c_i})$ for each candidate. These embeddings encapsulate the features and relationships pertinent to their respective texts and graphs. The cosine similarity between the embeddings of the query and each candidate is calculated and is then used for the selection of the most relevant options.

## 3 GT2Vec

In this work, we propose a simple yet effective framework GT2Vec to learn joint embeddings of text and graphs, as illustrated in Figure 1. Specifically, graph-structured data are first extracted into a graph token, which is then fed into the LLM backbone together with the text tokens (§3.1, §3.2). The LLM backbone outputs the joint embeddings, which can be used for the downstream tasks, such as classification and retrieval.

A key aspect of GT2Vec is the explicit alignment between the graph and text embeddings. This alignment is crucial, as it allows the LLM backbone to better integrate the structured knowledge from the graph with the unstructured text, improving the quality of the joint representations (§4). To achieve this, we introduce a contrastive learning mechanism that explicitly maps embeddings from both modalities into a shared space (§3.3).

While LLMs are commonly used for generation tasks in an autoregressive paradigm, GT2Vec takes a different route by utilizing LLMs as powerful encoders. This allows us to directly obtain robust joint embeddings of text and graph data by leveraging the rich contextual understanding of LLMs.

Our proposed architecture, GT2Vec, consists of three main components: a graph encoder, an MLP adapter, and a large language model backbone. Below, we discuss the components' design and implementation details, along with the alignment mechanism.

### 3.1 Graph Data Encoding

GT2Vec employs a graph encoder to obtain graph embeddings, which encapsulate essential information extracted from the contextual structure of the graph. Next, we describe the two-step process, which includes: (1) the integration of the query node in the graph; and (2) the graph encoding.

*3.1.1 Query Node Integration into Graph Structures.* We first initialize node embeddings within the input graph $\mathcal{G}$ with a language model (e.g., RoBERTa [38]). Following prior work [33, 64, 68], we link entities mentioned in the query to nodes in the graph, denoting these nodes as $\mathcal{V}_{\text{linked}}$. We then introduce a new node termed the query node, denoted as $v_q$. This query node is initialized using a language model by encoding the input query text. The query node

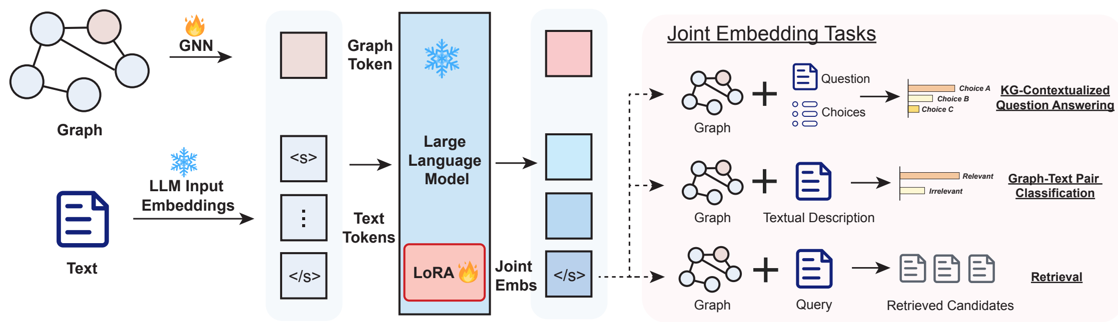


**Figure 1: Overview of GT2Vec framework. Unlike the common use of LLMs for generation tasks, we leverage LLMs to obtain joint embeddings of both text and graph data. We encode the input graph with a GNN, which provides the graph embeddings. The graph embeddings are then transformed into the word embedding space in the large language model. These embeddings are then fed into a large language model, and the outputs are utilized for various downstream tasks.**

$v_q$ is then connected to all nodes within $\mathcal{V}_{\text{linked}}$, enhancing the connection between the query and the nodes within the graph. We denote the updated graph with $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$.

*3.1.2 Graph Encoding Process.* The updated graph $\mathcal{G}'$ is then fed into an encoder for feature extraction. We adopt a modified version of graph attention network (GAT) [57, 64, 68] as the graph encoder. Specifically, in each layer of GAT, the message-passing process is formulated as

$$\boldsymbol{h}_v^{(\ell+1)} = f_n\left(\textstyle\sum_{s\in\mathcal{N}_v\cup\{v\}} \alpha_{sv}\mathbf{m}_{sv}\right) + \boldsymbol{h}_v^{(\ell)} \tag{1}$$

where $\mathcal{N}_v$ denotes the neighbors of node $v$, $\mathbf{m}_{sv}$ represents the message from each neighbor node $s$ to node $v$. $\alpha_{sv}$ is the attention weight. $f_n$ is a 2-layer MLP. The messages $\mathbf{m}_{sv}$ from node $s$ to $v$ are computed as the following:

$$\boldsymbol{r}_{sv} = f_r(\boldsymbol{e}_{sv}, \boldsymbol{u}_s, \boldsymbol{u}_v)\ \mathbf{m}_{sv} = f_m(\boldsymbol{h}_v^{(\ell)}, \boldsymbol{u}_v, \boldsymbol{r}_{sv}) \tag{2}$$

where $\boldsymbol{u}_s, \boldsymbol{u}_v$ denotes node type embeddings, and $\boldsymbol{e}_{sv}$ is edge embeddings, $f_r$ is a 2-layer MLP, and $f_m$ is a linear projection. Additionally, the attention weight $\alpha_{sv}$, which measures the importance of each neighbor's message, is calculated in the following manner:

$$\boldsymbol{q}_s = f_q(\boldsymbol{h}_s^{(\ell)}, \boldsymbol{u}_s)\ \boldsymbol{k}_v = f_k(\boldsymbol{h}_v^{(\ell)}, \boldsymbol{u}_v, \boldsymbol{r}_{sv}) \tag{3}$$

$$\gamma_{sv} = \frac{\boldsymbol{q}_s^\top \boldsymbol{k}_v}{\sqrt{D}}\ \alpha_{sv} = \frac{\exp(\gamma_{sv})}{\sum_{v'\in\mathcal{N}_s\cup\{s\}} \exp(\gamma_{sv'})} \tag{4}$$

where $f_q$ and $f_k$ are linear transformation functions, and $D$ is the hidden dimension. Following $L$ layers of message passing, we concatenate the final layer embeddings of query node $v_q$, the average pooling of the node embeddings in the final layer, and the text embeddings of the input query, then employ an MLP to generate the graph embeddings $\boldsymbol{g}$.

## 3.2 Fusion of Graph and Textual Data in LLM

We then integrate the graph embeddings with textual information using the LLM to produce embeddings suitable for downstream tasks. To facilitate this integration, we draw inspiration from practices in computer vision [37, 71], where image embeddings are first transformed into the text space to be processed by LLMs. Similarly, we also convert the graph embeddings into the text space. We employ an MLP adapter for this purpose, which projects the graph token embeddings into the language model space using a two-layer MLP with ReLU activation functions, leading to the transformed graph embeddings $\tilde{\boldsymbol{g}}$. The transformation allows us to insert the processed graph token at the beginning of the text sequence, formatted as [*graph token*, <*s*>, *token* 1, *token* 2, ..., </*s*>]. In this sequence, the graph token, output by the MLP adapter, precedes the textual tokens, which are derived from the initial input embeddings of the LLM. This arrangement ensures that the initial context for the LLM processing includes both graph-derived and textual information.

We then feed this sequence into the LLM. The embeddings of the </*s*> token from the last layer of the LLM are considered as the final output embeddings $\boldsymbol{z}$, encapsulating the combined knowledge of the graph and text inputs. This integration process leverages the LLM's capacity to synthesize information across different modalities, optimizing the embeddings for subsequent applications.

## 3.3 Graph-Text Alignment via Contrastive Learning

As we mentioned above, in GT2Vec, graph embeddings are converted into the text space via a MLP module and then are processed by the LLM together with the input text tokens. To better align the graph embedding space and text space, we consider contrastive learning for the graph-text alignment.

Contrastive learning has been broadly used in various domains [3, 16, 22, 34, 35, 62, 66]. The key idea behind contrastive learning is to minimize the distance between similar or positive pairs in the embedding space, while maximizing the separation between dissimilar or negative pairs. Although similar techniques have been applied in graph representation learning [10, 65, 66] and language models [9, 15, 23], these approaches typically operate within a single modality (either graphs or text). In contrast, our GT2Vec framework aligns the embedding spaces between text and graphs by (1) converting graph data into textual descriptions and (2) using

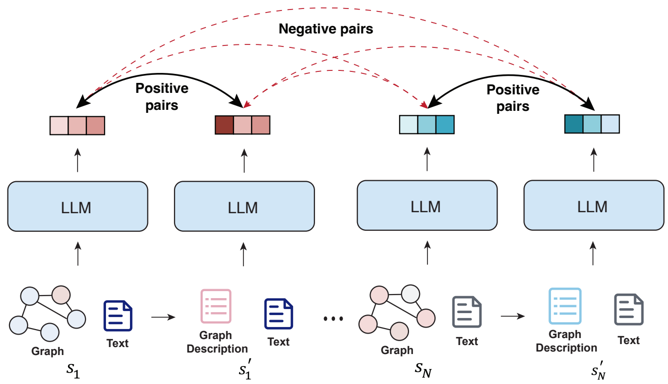


**Figure 2: Overview of graph-text alignment through contrastive learning.**

contrastive learning to strengthen the connections between the textual descriptions of graphs and their corresponding original graph representations, as shown in Figure 2.

*3.3.1 Translating Graphs into Textual Descriptions.* We first align the graph embeddings with text embeddings derived from descriptions of the same graph. Specifically, we start by listing all triples within the graph in a simple format: [entity (node), relation (edge), entity (node)]. This textual representation of the graph is then converted into a textual format by reorganizing the entities and relations into natural language. For example, the triples "*(analyzing, causes, new knowledge), (knowledge, causes, learn), (learn, causes, find information)*" are transformed into the following textual description: "*analyzing causes new knowledge; knowledge causes learn; learn new causes find information.*"

*3.3.2 Contrastive Learning for Alignment.* Now, we have two branches in the GT2Vec framework: the original graph + text branch and the graph description + text branch. These branches create a dual-view architecture that facilitates the alignment of graph and text embedding spaces through contrastive learning.

In the contrastive learning stage, we define positive pairs as the embeddings from the branch processing the original graph and its corresponding textual description that accurately reflect the same information. Negative pairs, on the other hand, consist of unrelated graph-text pairs from within the same batch, ensuring that the model learns to distinguish between semantically aligned and unaligned graph-text pair data. Specifically, for the original graph + text branch, we use the way mentioned in §3.2 to generate the joint embeddings $z_{\text{orig}}$. This involves combining graph embeddings, obtained from a graph encoder, with textual embeddings generated by the LLM's embedding layer to produce joint embeddings. For the graph description + text branch, we treat the input as pure text and use only the LLM to process it and generate the joint embeddings $z_{\text{new}}$. By applying infoNCE loss [45], we have

$$\mathcal{L}^{(\text{infoNCE})} = -\sum_{i=1}^{n_{\text{batch}}} \log\left(\frac{\exp\left(\tilde{z}_{\text{orig}}^{\text{T}}(i)\cdot\tilde{z}_{\text{new}}(i)\right)}{\sum_{j=1}^{n_{\text{batch}}}\exp\left(\tilde{z}_{\text{orig}}^{\text{T}}(i)\cdot\tilde{z}_{\text{new}}(j)\right)}\right) \tag{5}$$

where $n_{\text{batch}}$ represents the number of samples in a training batch, and $\tilde{z}_{\text{orig}}$ and $\tilde{z}_{\text{new}}$ denote the normalized vectors of $z_{\text{orig}}$ and $z_{\text{new}}$, respectively. As we present in §4, the contrastive learning method improves GT2Vec's ability to better align the graph and text embeddings, which is further beneficial to the downstream tasks.

## 3.4 Training

Our proposed framework, GT2Vec, can accommodate a wide array of tasks that require the integration of graph data and text. The training process for each task incorporates a task-specific loss function combined with a contrastive learning loss. The general form of the combined loss for each task can be represented as follows:

$$\mathcal{L} = \mathcal{L}^{(\text{task})} + \lambda\mathcal{L}^{(\text{infoNCE})} \tag{6}$$

where $\lambda$ is a hyperparameter used to adjust the weights between loss functions. $\mathcal{L}^{(\text{task})}$ refers to the loss directly associated with the primary objective of the task. For KG-Contextualized QA, we use cross-entropy loss; for graph-text pair classification, we apply binary cross-entropy (BCE) loss; and for retrieval, we use infoNCE loss. More details can be found in the Appendix C.2.

# 4 Experiments

Next, we assess the performance of GT2Vec and compare it to strong baselines on three types of tasks: KG-contextualized QA, graph-text pair classification, and retrieval tasks (§4.1-4.3). We also perform extensive ablation studies to understand the impact of our design and other choices (§4.4).

## 4.1 KG-contextualized QA Performance

*4.1.1 Datasets and Metrics.* We first evaluated GT2Vec on three QA datasets, i.e., CommonsenseQA [53], OpenBookQA [41], and MedQA-USMLE [26]. We split these datasets according to Yasunaga et al. [64] and Zhang et al. [68] into train, validation, and test splits. For evaluation, we use accuracy as the metric to measure the performance on each dataset. For each question in the QA datasets, a subgraph context extracted from a KG is utilized to provide additional contextual information, following Yasunaga et al. [64] (see Appendix B.1).

*4.1.2 Baselines.* We compare GT2Vec with a range of baseline models, including both language models (LM) and hybrid approaches that integrate language models with knowledge graphs (LM+KG).

**Fine-tuned LMs.** We consider the following vanilla fine-tuned language models (LMs): RoBERTa-Large [38] and E5-Mistral [60]. Additionally, for MedQA-USMLE dataset, we use several domain-specific models, including SapBERT [36] and BioBERT [29].

**Existing LM+KG models.** Our LM+KG baselines include: GreaseLM [68] QAGNN [64], Relation Network (RN) [50], RGCN [51], GconAttn [61], and MHGRN [11]. Unlike these models that leverage GNNs as the backbone, our framework GT2Vec adopts more powerful LLMs for better performance. We adopt E5-Mistral [60] as GT2Vec's LLM backbone.

*4.1.3 Training Details.* During training, we adopt parameter efficient finetuning techniques, including Linear Probe (LP) [48] and Low-Rank Adaptation (LoRA) [19], for E5-Mistral and GT2Vec. LP freezes the pre-trained LLM and trains a linear classifier on top, while LoRA introduces low-rank adaptation layers for efficient fine-tuning. we employ full parameter fine-tuning for the other baselines.

**Table 1: Test accuracy comparison on CommonsenseQA and OpenBookQA. The baseline results are mainly sourced from Zhang et al. [68] and Yasunaga et al. [64]. Bold indicates the best result, and underline indicates the second best. LP means linear probe.**

| Methods | CommonsenseQA | OpenBookQA (w/o Scientific Facts) | OpenBookQA (w/ Scientific Facts) |
|---|---|---|---|
| ***Language Models Only*** | | | |
| RoBERTa-Large [38] | 68.69 ± 0.56 | 64.80 ± 2.37 | 78.40 ± 1.64 |
| E5-Mistral, LP [60] | 69.49 ± 0.28 | 74.80 ± 0.35 | 81.67 ± 0.31 |
| E5-Mistral, LoRA [60] | 78.73 ± 0.16 | 85.60 ± 0.20 | 91.87 ± 0.12 |
| ***LM + KG*** | | | |
| RGCN [51] | 68.41 ± 0.66 | 62.45 ± 1.57 | 74.60 ± 2.53 |
| GconAttn [61] | 68.59 ± 0.39 | 64.75 ± 1.48 | 71.80 ± 1.21 |
| MHGRN [11] | 71.11 ± 0.10 | 66.85 ± 1.19 | 81.87 ± 1.86 |
| QA-GNN [64] | 73.41 ± 0.92 | 67.80 ± 2.75 | 82.77 ± 1.56 |
| GreaseLM [68] | 74.20 ± 0.40 | 65.60 ± 0.40 | 83.87 ± 1.29 |
| GT2Vec, LP (Ours) | <u>81.09</u> ± 0.73 | <u>86.67</u> ± 1.10 | <u>93.33</u> ± 0.42 |
| GT2Vec, LoRA (Ours) | **81.39** ± 0.11 | **88.13** ± 0.42 | **93.67** ± 0.31 |

**Table 2: Test accuracy comparison on MedQA-USMLE. The baseline results are mainly sourced from Zhang et al. [68]. Bold indicates the best result, and underline indicates the second best. LP means linear probe.**

| Methods | Test Acc |
|---|---|
| ***Language Models Only*** | |
| BERT-Base [7] | 34.3 |
| BioBERT-Base [29] | 34.1 |
| RoBERTa-Large [38] | 35 |
| BioBERT-Large [29] | 36.7 |
| SapBERT [36] | 37.2 |
| E5-Mistral, LP [60] | 39.4 ± 1.1 |
| E5-Mistral, LoRA [60] | <u>51.1</u> ± 0.3 |
| ***LM + KG*** | |
| QA-GNN [64] | 38 |
| GreaseLM [68] | 38.5 |
| GT2Vec, LP (Ours) | 49.9 ± 0.9 |
| GT2Vec, LoRA (Ours) | **53.4** ± 0.3 |

**Table 3: Test accuracy comparison on WebNLG dataset.**

| Methods | Test Acc |
|---|---|
| ***LM + KG*** | |
| RGCN [51] | 63.20 ± 0.49 |
| MHGRN [11] | 84.98 ± 0.53 |
| QA-GNN [64] | 75.55 ± 3.54 |
| GreaseLM [68] | 82.50 ± 4.29 |
| GT2Vec, LP (Ours) | <u>88.43</u> ± 1.33 |
| GT2Vec, LoRA (Ours) | **89.70** ± 0.34 |

Further details on the training process such as hyperparameters can be found in Appendix D.

*4.1.4 Results.* We first conduct comparison experiments on CommonsenseQA and OpenBookQA datasets, as illustrated in Table 1. In both datasets, our framework outperforms all other methods significantly. Specifically, on CommonsenseQA, it surpasses the strongest baseline E5-Mistral, achieving a test accuracy of 81.39%. On OpenBookQA, GT2Vec also demonstrates superior performance, achieving 88.13% test accuracy. When further integrating the extra corpus of scientific facts provided by OpenbookQA [5], our model reaches a remarkable 93.67% accuracy, outperforming all the baseline models that also utilized scientific facts. The remarkable performance can be attributed primarily to the robust capabilities of the LLM backbone integrated within our framework. The LLM backbone not only enhances language understanding but also integrates the contextual knowledge derived from graphs, thereby substantially improving KG-contextualized QA performance.

We further evaluate GT2Vec on the MedQA-USMLE dataset to assess its generalization capability across specialized domains and report the results in Table 2. GT2Vec achieves a test accuracy of 53.4%, outperforming all the baselines including domain-specific models. This result further reinforces the adaptability of GT2Vec across different domains showcasing its ability to excel not only in general and scientific question answering but also in knowledge-intensive and highly specialized fields.

## 4.2 Graph-Text Pair Classification Performance

Building upon the strong results achieved in the KG-contextualized QA tasks, we further evaluate GT2Vec in a different task: graph-text pair classification. Our objective in evaluating this task is to assess GT2Vec's ability to generalize beyond QA scenarios, demonstrating its versatility in handling a broader range of graph-text embedding tasks. We evaluate models on the WebNLG [14] dataset and use accuracy as the evaluation metric, measuring the proportion of correctly classified graph-text pairs.

**Table 4: Comparision results (NDCG@10) on retrieval tasks.**

| Methods/Datasets | SciFact | FIQA |
|---|---|---|
| ***Language Models Only*** | | |
| BERT-Base [7] | 13.3 | 2.2 |
| RoBERTa-Large [38] | 43.3 | 20.4 |
| E5-Small [59] | 65.6 | 34.8 |
| E5-Base [59] | 73.1 | 36.4 |
| E5-Large [59] | 72.6 | 38.6 |
| GTR-XXL [43] | 66.2 | 46.7 |
| SGPT [42] | 74.7 | 37.2 |
| E5-Mistral, 0-shot [60] | 76.1 | 53.5 |
| E5-Mistral, LoRA [60] | 75.9 | 53.2 |
| ***LM + KG*** | | |
| QA-GNN [64] | 41.4 | 19.5 |
| GreaseLM [68] | 48.9 | 29.3 |
| GT2Vec, LP (Ours) | **82.9** | 54.1 |
| GT2Vec, LoRA (Ours) | 80.8 | **56.2** |

The results of the graph-text pair classification task are presented in Table 3. GT2Vec significantly outperforms all baseline models by at least 4.72%, achieving a test accuracy of 89.70%. While models like GreaseLM and MHGRN perform well by incorporating knowledge graphs, they lack the deeper contextual understanding due to the limitations of their less powerful backbone models, such as shallow transformers or GNNs.

## 4.3 Retrieval Performance

Next, we further evaluate GT2Vec's performance on retrieval tasks, where the objective is to retrieve relevant sentences or documents based on a query with graph-context. We conduct experiments on two datasets, SciFact [58] and FiQA [40], and report NDCG@10 as the primary evaluation metric.

As shown in Table 4, GT2Vec achieves the best performance on both datasets. A closer examination of the LM + KG baselines reveals interesting trends. Both models use RoBERTa-Large in their architectures. From the results, GreaseLM outperforms the RoBERTa-Large model by a margin of 5.6% on SciFact and 8.9% on FiQA, which aligns with our earlier findings that graph context can be useful for enhancing retrieval performance. However, QA-GNN shows a performance degradation compared to RoBERTa-Large, particularly on FiQA (19.5 vs. 20.4). The reason for this drop may be attributed to QA-GNN's use of a simple MLP for combining the graph and text embeddings. This weaker integration mechanism is likely insufficient for fully leveraging the graph context, resulting in suboptimal performance. In contrast, GreaseLM employs a shallow transformer, which offers better capacity for fusing multimodal information, leading to moderate gains.

## 4.4 Ablation Studies

To better understand the contributions of different components in GT2Vec, we perform an ablation study on the CommonsenseQA and OpenBookQA datasets under the linear probe setting. More results can be found in Appendix D.2.

**Effect of LLM Backbone Choice.** GT2Vec exhibits flexibility in adopting various LLMs as its backbone. To evaluate the impact of different LLM backbones on performance, we compare four series of models: the E5 series [59, 60], LLaMA-2 [55], LLaMA-3 [8], and Mistral [24], as illustrated in Figure 3(a). We first observe that increasing the model size within each series consistently enhances the performance. This trend highlights that larger models, with more parameters, are typically able to better capture complex relationships in multi-modal data, leading to higher performance. Additionally, LLaMA-3 significantly improves over LLaMA-2. Specifically, the performance of LLaMA-3-3B matches that of LLaMA-2-7B and LLaMA-3-8B achieves results comparable to LLaMA-2-13B, indicating that the new LLaMA-3 architecture brings considerable advancements compared to its predecessor LLaMA-2.

**Effect of Graph Encoder Depth.** We evaluate the effect of varying the number of GAT layers in the graph encoder on the performance of GT2Vec using the CommonsenseQA and OpenBookQA datasets. As shown in Figure 3(b), both datasets exhibit slight fluctuations in test accuracy as the number of GAT layers increases from 3 to 7. This suggests that the method is quite robust to changes in GAT depth, with only small variations in performance.

**Effect of Graph-Text Alignment.** We further investigate the effectiveness of graph-text alignment by studying how the alignment evolves during training. Specifically, we calculate the mean Euclidean distance between the normalized graph and text embeddings on the dev set (shown as the red dashed line) and compare it to the corresponding development accuracy (purple solid line) across different training epochs, as illustrated in Figure 3(c). This figure shows a clear inverse relationship between the two curves: as the graph-text distance decreases, accuracy improves. Notably, at epoch 2, the graph-text distance reaches its minimum, while the dev accuracy peaks. This trend suggests that better alignment between graph and text embeddings contributes directly to improved model performance, highlighting the effectiveness of our graph-text alignment strategy in GT2Vec.

# 5 Related Work

**Graphs and Language Models for Multi-Modal Embedding Tasks.** Integration of graph data with language models has been an evolving field, aiming to combine structured graph data with unstructured textual information for enhanced data analysis. Historically, early attempts in this area employed an MLP or a shallow transformer to merge the information from both modalities [11, 33, 64, 68], which may not fully exploit the rich contextual information from text and graph data.

The advent of LLMs has brought a transformative shift to the NLP field. LLMs, with their extensive pre-training on diverse corpora, offer unprecedented capabilities for deep semantic understanding and reasoning [13, 70]. Recent research has begun to explore the potential of LLMs for enhancing multimodal embedding tasks of graphs and text [17, 21, 54, 69, 73]. These studies primarily focus on augmenting the graph representation capabilities within the LLM framework. While these efforts mark significant advancements, they predominantly concentrate on graph representation learning rather than direct NLP tasks. Additionally, these methods do not explicitly align the semantic spaces of graphs and text. In

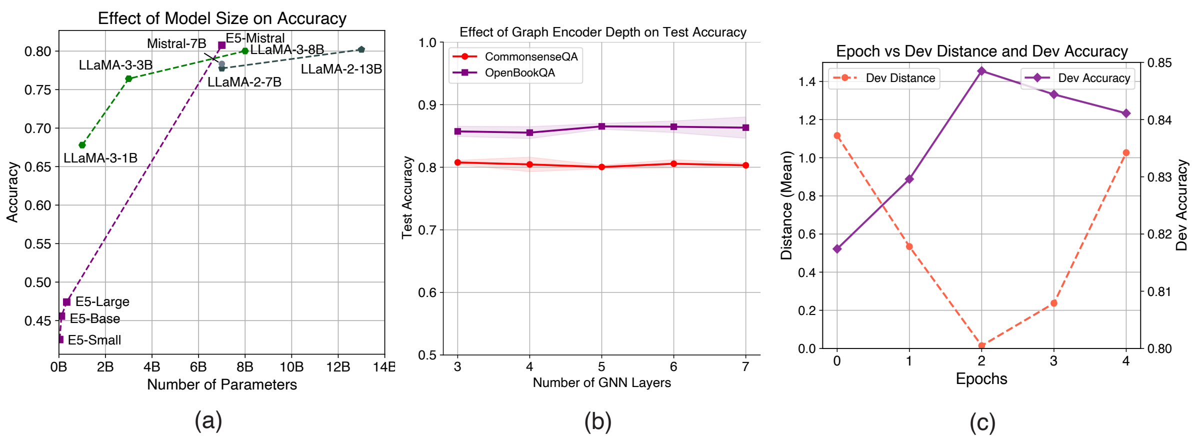


**Figure 3: (a) The effect of LLM backbone choice on accuracy for the CommonsenseQA dataset. The figure shows three series: E5, LLaMA-3, and LLaMA-2, along with a single Mistral-7B model. (b) The effect of graph encoder depth (number of GNN layers) on test accuracy for CommonsenseQA and OpenBookQA datasets. The shaded areas represent the standard deviation, indicating the variance in performance across different trials. (c) Graph-text embedding distance (red dashed) and dev accuracy (purple solid) on CommonsenseQA across training epochs.**

contrast, our approach not only integrates LLMs for handling complex multimodal inputs but also introduces an explicit alignment mechanism between graph and text embeddings. This alignment is crucial for enhancing the semantic integration and boosting performance across diverse NLP tasks by capturing complex intermodal relationships.

There are recent studies that have explored using LLMs and graph data for generative tasks beyond embedding tasks [1, 18, 20, 25, 47, 67, 72]. Our work, however, concentrates on embedding tasks, which is an orthogonal direction. Unlike generative tasks, which primarily focus on creating new content via next-token prediction, embedding tasks are aimed at developing rich, informative representations that can be used directly to enhance performance in downstream applications such as classification and retrieval.

**Contrastive Learning.** Contrastive learning has gained significant attention in recent years as a method to learn representations by maximizing agreement between positive pairs while pushing negative pairs apart in the embedding space [3, 16, 34, 35, 66]. Pioneering works like SimCLR [3] and MoCo [16] have demonstrated the effectiveness of contrastive learning in the visual domain. Similar approaches have been applied to graphs [10, 65, 66] and language models [9, 15, 23], but these methods typically operate within a single modality (e.g., graphs or text). In contrast, our approach introduces contrastive learning to align the graph embedding space with the text embedding space. By transforming graphs into textual descriptions and applying contrastive learning, we ensure that embeddings from both modalities align closely, enhancing the model's ability to perform tasks that require both graph-based reasoning and text comprehension.

# 6 Conclusion

In this paper, we introduce GT2Vec, a simple yet effective framework designed to integrate graph and text data using a novel alignment strategy and LLMs. We have demonstrated that GT2Vec enhances the semantic coherence between these two modalities, resulting in significantly improved performance on several NLP tasks and datasets. By aligning graph embeddings directly with text embeddings, GT2Vec ensures a deeper integration of structured and unstructured data.

# References

[1] Ziwei Chai, Tianjie Zhang, Liang Wu, Kaiqiao Han, Xiaohai Hu, Xuanwen Huang, and Yang Yang. 2023. Graphllm: Boosting graph reasoning ability of large language model. *arXiv preprint arXiv:2310.05845* (2023).

[2] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. MiniGPT-v2: large language model as a unified interface for vision-language multi-task learning. *CoRR* abs/2310.09478 (2023). https://doi.org/10.48550/ARXIV.2310.09478 arXiv:2310.09478

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[4] Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon. 2021. Evaluation of bert and albert sentence embedding performance on downstream nlp tasks. In *2020 25th International conference on pattern recognition (ICPR)*. IEEE, 5482–5487.

[5] Peter Clark, Oren Etzioni, Tushar Khot, Daniel Khashabi, Bhavana Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, et al. 2020. From 'F'to 'A'on the NY regents science exams: An overview of the aristo project. *Ai Magazine* 41, 4 (2020), 39–53.

[6] Anthony M. Colas, Mehrdad Alvandipour, and Daisy Zhe Wang. 2022. GAP: A Graph-aware Language Model Framework for Knowledge Graph-to-Text Generation. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na (Eds.). International Committee on Computational Linguistics, 5755–5769. https://aclanthology.org/2022.coling-1.506

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill

Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/V1/N19-1423

[8] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[9] Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766* (2020).

[10] Yi Fang, Dongzhe Fan, Daochen Zha, and Qiaoyu Tan. 2024. Gaugllm: Improving graph contrastive learning for text-attributed graphs with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 747–758.

[11] Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable Multi-Hop Relational Reasoning for Knowledge-Aware Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 1295–1309. https://doi.org/10.18653/V1/2020.EMNLP-MAIN.99

[12] Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428* (2019).

[13] Kaiyuan Gao, Sunan He, Zhenyu He, Jiacheng Lin, QiZhi Pei, Jie Shao, and Wei Zhang. 2023. Examining user-friendly and open-sourced large gpt models: A survey on language, multimodal, and scientific gpt models. *arXiv preprint arXiv:2308.14149* (2023).

[14] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating Training Corpora for NLG Micro-Planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, 179–188. https://doi.org/10.18653/v1/P17-1017

[15] John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 879–895.

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

[17] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2024. Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=RXFVcynVe1

[18] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *arXiv preprint arXiv:2402.07630* (2024).

[19] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=nZeVKeeFYf9

[20] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024. GRAG: Graph Retrieval-Augmented Generation. *arXiv preprint arXiv:2405.16506* (2024).

[21] Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. 2024. GNNs as Adapters for LLMs on Text-Attributed Graphs. In *The Web Conference 2024*.

[22] Dan Iter, Kelvin Guu, Larry Lansing, and Dan Jurafsky. 2020. Pretraining with Contrastive Sentence Objectives Improves Discourse Performance of Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 4859–4870. https://doi.org/10.18653/V1/2020.ACL-MAIN.439

[23] Dan Iter, Kelvin Guu, Larry Lansing, and Dan Jurafsky. 2020. Pretraining with Contrastive Sentence Objectives Improves Discourse Performance of Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4859–4870.

[24] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* (2023).

[25] Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, and Jiawei Han. 2024. Graph Chain-of-Thought: Augmenting Large Language Models by Reasoning on Graphs. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 163–184. https://aclanthology.org/2024.findings-acl.11

[26] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What Disease does this Patient Have? A Large-scale Open Domain Question Answering Dataset from Medical Exams. *CoRR* abs/2009.13081 (2020). arXiv:2009.13081 https://arxiv.org/abs/2009.13081

[27] Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. JointGT: Graph-Text Joint Representation Learning for Text Generation from Knowledge Graphs. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021 (Findings of ACL, Vol. ACL/IJCNLP 2021)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 2526–2538. https://doi.org/10.18653/V1/2021.FINDINGS-ACL.223

[28] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

[29] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.

[30] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195. https://doi.org/10.3233/SW-140134

[31] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 19730–19742. https://proceedings.mlr.press/v202/li23q.html

[32] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 2829–2839. https://doi.org/10.18653/V1/D19-1282

[33] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2829–2839.

[34] Jiacheng Lin, Hanwen Xu, Addie Woicik, and Jianzhu Ma. [n. d.]. Pisces: A Cross-Modal Contrastive Learning Approach to Synergistic Drug Combination Prediction. In *Research in Computational Molecular Biology*. Springer, 268.

[35] Jiacheng Lin, Meng Xu, Zhihua Xiong, and Huangang Wang. 2024. CAMBranch: Contrastive Learning with Augmented MILPs for Branching. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=K6kt50zAiG

[36] Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. Self-Alignment Pretraining for Biomedical Entity Representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, 4228–4238. https://doi.org/10.18653/V1/2021.NAACL-MAIN.334

[37] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html

[38] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[39] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-Tuning LLaMA for Multi-Stage Text Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (Eds.). ACM, 2421–2425. https://doi.org/10.1145/3626772.3657951

[40] Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. WWW'18 Open Challenge: Financial Opinion Mining and Question Answering. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 1941–1942. https://doi.org/10.1145/3184558.3192301

[41] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, 2381–2391. https://doi.org/10.18653/V1/D18-1260

[42] Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904* (2022).

[43] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, et al. 2022. Large Dual Encoders Are Generalizable Retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 9844–9855.

[44] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large Dual Encoders Are Generalizable Retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 9844–9855. https://doi.org/10.18653/V1/2022.EMNLP-MAIN.669

[45] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[47] Bryan Perozzi, Bahare Fatemi, Dustin Zelle, Anton Tsitsulin, Mehran Kazemi, Rami Al-Rfou, and Jonathan Halcrow. 2024. Let your graph do the talking: Encoding structured data for llms. *arXiv preprint arXiv:2402.05862* (2024).

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[49] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. https://doi.org/10.18653/V1/D19-1410

[50] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Tim Lillicrap. 2017. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 4967–4976. https://proceedings.neurips.cc/paper/2017/hash/e6acf4b0f69f6f6e60e9a815938aa1ff-Abstract.html

[51] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings (Lecture Notes in Computer Science, Vol. 10843)*, Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam (Eds.). Springer, 593–607. https://doi.org/10.1007/978-3-319-93417-4_38

[52] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, Satinder Singh and Shaul Markovitch (Eds.). AAAI Press, 4444–4451. https://doi.org/10.1609/AAAI.V31I1.11164

[53] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4149–4158. https://doi.org/10.18653/V1/N19-1421

[54] Yijun Tian, Huan Song, Zichen Wang, Haozhu Wang, Ziqing Hu, Fang Wang, Nitesh V Chawla, and Panpan Xu. 2024. Graph neural prompting with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19080–19088.

[55] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[57] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rJXMpikCZ

[58] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 7534–7550. https://doi.org/10.18653/V1/2020.EMNLP-MAIN.609

[59] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533* (2022).

[60] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368* (2023).

[61] Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, and Michael Witbrock. 2019. Improving Natural Language Inference Using External Knowledge in the Science Questions Domain. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 7208–7215. https://doi.org/10.1609/AAAI.V33I01.33017208

[62] H Xu, J Lin, A Woicik, Z Liu, J Ma, S Zhang, H Poon, L Wang, and S Wang. 2022. Pisces: A multi-modal data augmentation approach for drug combination synergy prediction. (2022).

[63] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=ryGs6iA5Km

[64] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, 535–546. https://doi.org/10.18653/V1/2021.NAACL-MAIN.45

[65] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems* 33 (2020), 5812–5823.

[66] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1294–1303.

[67] Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng Yang, and Chuan Shi. 2024. GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks. In *Proceedings of the ACM on Web Conference 2024*. 1003–1014.

[68] Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022. GreaseLM: Graph REASoning Enhanced Language Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=41e9o6cQPj

[69] Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. 2023. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089* (2023).

[70] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).

[71] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. In *The Twelfth International Conference on Learning*

*Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=1tZbq88f27

[72] Kerui Zhu, Bo-Wei Huang, Bowen Jin, Yizhu Jiao, Ming Zhong, Kevin Chang, Shou-De Lin, and Jiawei Han. 2024. Investigating Instruction Tuning Large Language Models on Graphs. In *First Conference on Language Modeling*. https://openreview.net/forum?id=xdg4CS5mkl

[73] Qi Zhu, Da Zheng, Xiang Song, Shichang Zhang, Bowen Jin, Yizhou Sun, and George Karypis. 2024. Parameter-Efficient Tuning Large Language Models for Graph Representation Learning. *arXiv preprint arXiv:2404.18271* (2024).

# A Ethical Use of Data and Informed Consent

All datasets used in this work are publicly available and widely used in the research community. No personally identifiable information (PII) or sensitive data is included in these datasets. Our research strictly adheres to the ethical guidelines of using publicly available datasets, and no additional data was collected from human subjects.

# B Dataset Details

## B.1 KG-Contextulized QA

**CommonsenseQA** [53] is a 5-way multiple-choice QA dataset focused on applying commonsense knowledge in answering questions. It includes 12,102 questions, each with one correct answer and four distractor answers.

**OpenBookQA** [41] is a 4-way multiple choice QA dataset that requires reasoning with elementary science knowledge, containing 5,957 questions.

**MedQA-USMLE** [26] is a 4-way multiple choice QA task that requires biomedical and clinical knowledge. The questions are originally from practice tests for the United States Medical License Exams (USMLE). The dataset contains 12,723 questions.

For each question in the QA datasets, a subgraph context extracted from a KG is utilized to provide additional contextual information, following Yasunaga et al. [64]. Specifically, for CommonsenseQA and OpeBookQA, we used the ConceptNet [52], which contains 799,273 nodes and 2,487,810 edges. Node embeddings are initialized by Roberta-Large [38] and kept frozen during the training process, following Yasunaga et al. [64]. The query node embeddings mentioned in Section 3.1 are calculated by the LLM backbone. For MedQA-USMLE dataset, we used the UMLS knowledge graph used in Yasunaga et al. [64], which contains 9,958 nodes and 44,561 edges. Node embeddings are initialized using SapBERT [36], following Yasunaga et al. [64].

## B.2 Graph-Text Pair Classification

To evaluate GT2Vec on graph-text pair classification, we curated a dataset based on **WebNLG** [14]. The original WebNLG dataset is used to assess models' ability in text-to-graph and graph-to-text generation. Concretely, each data contains a graph-text pair where the graph is a set of triples from DBpedia and the text is the corresponding description of the triples. For example, the graph data are *"(John_E_Blaha birthDate 1942_08_26), (John_E_Blaha birthPlace San_Antonio), (John_E_Blaha occupation Fighter_pilot)"*, while the corresponding text is *"John E Blaha, born in San Antonio on 1942-08-26, worked as a fighter pilot."*

In this paper, we use the v3.0 release data to construct the dataset for graph-text pair classification. First, we curated the dataset by identifying relations that appear in the training, dev, and test sets. We then filtered the dataset to retain only those graph triples that contain these relations for all three sets. To further increase the complexity of the task, we generated a series of new, randomly combined positive samples. Specifically, we merged multiple graph-text pairs from the original dataset, creating data with larger graphs and longer text by concatenating their respective triples and sentences.

Next, we generated an equal number of negative samples. To this end, we followed a similar process to create positive pairs but introduced mismatches. In this case, while the graph triples were taken from one pair, the text was taken from a different, unrelated pair. This ensured that the graph and text did not align, resulting in non-matching pairs that serve as negative examples. We finally constructed 10,000 training, 4,000 validation, and 2,000 test data.

## B.3 Retrieval

**SciFact** [58] is a scientific fact-checking dataset aimed at verifying scientific claims using relevant research papers. It contains 920 training queries, and 300 test queries, with a corpus of 5,183 documents. In our experiment, we split 100 samples from the training set to serve as validation queries, ensuring that the remaining training data is used for model training while still providing a separate validation set for tuning and evaluation.

**FiQA** [40] is a financial-domain retrieval dataset designed to address complex financial question answering and information retrieval tasks. It contains 14,166 training queries, 500 development queries, and 648 test queries, with a total of 57,638 documents in the corpus.

# C Methodology Details

## C.1 Additional Framework Details

GT2Vec utilizes a dual-view architecture for graph-text alignment, incorporating two parallel branches. The first branch processes the original graph together with the text, while the second branch processes a textual description of the graph along with the input text. In our experiments, both branches are employed to generate embeddings that are used for downstream tasks, and both contribute to the task-specific loss $\mathcal{L}^{(\text{task})}$. This dual-branch approach has several advantages: it allows the model to learn complementary information from both the structured graph and its textual description, enhancing the overall representation. During the evaluation, however, we simplify the process by using only the graph+text branch, and it has shown to provide sufficient performance for the downstream tasks, eliminating the need for the graph description branch.

## C.2 Training Objective

In this part, we detail the task-specific loss function $\mathcal{L}^{(\text{task})}$ for each downstream task.

*C.2.1 KG-Contextualized QA.* For the KG-contextualized QA task, we apply the cross-entropy loss function, which is crucial for classification tasks involving multiple-choice questions. This loss function is computed as follows:

$$\mathcal{L}^{(\text{task})} = -\sum_{i=1}^{n_{\text{batch}}} \sum_{j=1}^{n_{\text{choice}}} y_j^{(i)} \log(\hat{y}_j^{(i)}) \tag{7}$$

where $n_{\text{batch}}$ is the number of samples in a batch and $n_{\text{choice}}$ is the number of answer choices per question. $y_j^{(i)}$ is a binary indicator (1 if the choice $j$ is the correct answer for sample $i$, and 0 otherwise). $\hat{y}_j^{(i)}$ is the predicted probability that choice $j$ is the correct answer for sample $i$.

*C.2.2 Graph-Text Pair Classification Tasks.* In the graph-text pair classification task, we use binary cross-entropy loss to determine the match/mismatch status between the graph and text pairs:

$$\mathcal{L}^{(\text{task})} = -\sum_{i=1}^{n_{\text{batch}}} \left( y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)} \log(1 - \hat{y}^{(i)}) \right) \tag{8}$$

Here, $y^{(i)}$ is the true label (1 if the pair matches, 0 otherwise) and $\hat{y}^{(i)}$ is the predicted probability of a match as output by the MLP classifier.

*C.2.3 Retrieval Tasks.* For retrieval tasks, we apply infoNCE loss [45] for training. Specifically, given a batch of positive pairs $(d_i, p_i)$, we assume that $(d_i, p_i)$ is a positive pair and $(d_i, p_j)$ for $i \neq j$ a negative pair. By applying infoNCE loss, we have

$$\mathcal{L}^{(task)} = -\sum_{i=1}^{n_{\text{batch}}} \log \left( \frac{e^{\text{sim}(a_i, p_i)/\tau}}{e^{\text{sim}(a_i, p_i)/\tau} + \sum_{j \neq i} e^{\text{sim}(a_i, p_j)/\tau}} \right) \tag{9}$$

where $\text{sim}(\cdot, \cdot)$ is the consine similarity between the embeddings. $\tau$ is the temperature scaling parameter.

**Table 5: Ablation Study on different branches of GT2Vec: This table reports the test accuracy of GT2Vec when using only the graph + text branch or only the graph description + text branch.**

| Models | CommonsenseQA | OpenBookQA |
|---|---|---|
| GreaseLM [68] | 74.20 ± 0.40 | 83.87 ± 1.29 |
| Only graph + text | 79.69 ± 0.85 | 84.80 ± 1.06 |
| Only description + text | 78.32 ± 0.61 | 84.00 ± 0.92 |
| GT2Vec | **81.09** ± 0.73 | **86.67** ± 1.10 |

**Table 6: The effect of graph encoder choice on test accuracy.**

| Graph Encoder | CommonsenseQA | OpenBookQA |
|---|---|---|
| GCN [28] | 80.95 ± 0.28 | 84.67 ± 1.27 |
| GAT [57] | 81.09 ± 0.73 | 86.67 ± 1.10 |
| GIN [63] | 80.23 ± 0.57 | 85.40 ± 0.92 |

**Table 7: LoRA Hyperparameters**

| Hyperparameter | Value |
|---|---|
| Rank ($r$) | 8 |
| Alpha ($\alpha$) | 16 |
| Dropout | 0.05 |
| Target Modules | {q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj} |

# D Experiment Details

## D.1 Additional Implementation Details

We implement models using PyTorch [46] and PyTorch Geometric [12]. All experiments are conducted on a single A100 80GB GPU. For baselines that we implemented ourselves, we followed the settings

**Table 8: Hyperparameters for training GT2Vec.**

| Hyperparameter | CommonsenseQA | OpenBookQA | MedQA-USMLE | WebNLG | SciFact | FiQA |
|---|---|---|---|---|---|---|
| GNN hidden dim | 256 | 256 | 256 | 256 | 256 | 512 |
| Number of GNN layers | 3 | 5 | 5 | 5 | 3 | 3 |
| GAT attention heads | 2 | 2 | 2 | 2 | 2 | 2 |
| Dropout rate | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Context length | 128 | 128 | 256 | 256 | 512 | 512 |
| Learning rate | $1 \times 10^{-5}$ | $1 \times 10^{-5}$ | $1 \times 10^{-5}$ | $1 \times 10^{-4}$ | $1 \times 10^{-5}$ | $5 \times 10^{-6}$ |
| Optimizer | RAdam | RAdam | RAdam | RAdam | RAdam | RAdam |
| Weight decay | $1 \times 10^{-2}$ | $1 \times 10^{-2}$ | $1 \times 10^{-2}$ | $1 \times 10^{-2}$ | $1 \times 10^{-2}$ | $1 \times 10^{-2}$ |
| Learning rate schedule | constant | constant | constant | constant | constant | constant |
| Number of epochs | 5 | 5 | 15 | 5 | 15 | 1 |
| Batch size | 8 | 8 | 8 | 32 | 16 | 256 |
| Max gradient norm | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| $\lambda$ in Eq. (6) | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| Knowledge graphs | ConceptNet | ConceptNet | UMLS | DBPedia | ConceptNet | ConceptNet |
| Max number of nodes in subgraphs | 200 | 200 | 200 | 200 | 200 | 200 |
| Number of relations | 38 | 38 | 34 | 748 | 38 | 38 |

**Table 9: Ablation study results on test accuracy.**

| Methods | CommonsenseQA | OpenBookQA |
|---|---|---|
| GreaseLM [68] | 74.20 ± 0.40 | 83.87 ± 1.29 |
| GT2Vec w/o graph | 69.49 ± 0.28 | 74.80 ± 0.35 |
| GT2Vec w/o alignment | 79.69 ± 0.85 | 84.80 ± 1.06 |
| GT2Vec | **81.09** ± 0.73 | **86.67** ± 1.10 |

and hyperparameters described in the original papers to ensure fair comparisons. Detailed hyperparameter settings for GT2Vec can be found in Table 7 and Table 8.

## D.2 Additional Results

*D.2.1 Effect of graph context information.* When GT2Vec is input with only textual input, excluding graph context information, there is a significant drop in performance, with accuracy decreasing to 69.49% on CommonsenseQA and 74.80% on OpenBookQA. This sharp decline underscores the critical role that graph context plays in enhancing the model's ability to understand and reason with the data. The graph provides structured knowledge that complements the unstructured text, and without it, the model relies solely on the textual input, which limits its capacity to effectively handle complex embedding tasks.

*D.2.2 Effect of Graph-Text Alignment.* Removing contrastive learning from GT2Vec leads to a performance drop, with the accuracy decreasing from 81.09% to 79.69% on CommonsenseQA, and from 86.67% to 84.80% on OpenBookQA. This highlights the importance of contrastive learning in aligning graph and text embeddings, ensuring better integration of multimodal information.

*D.2.3 Effect of Different Branches in GT2Vec.* To better understand the contributions of each branch in GT2Vec, we perform an ablation study, as shown in Table 5. This analysis evaluates the performance of GT2Vec when using only the graph + text branch or only the graph description + text branch and compares it to the full version of GT2Vec, which integrates both branches. We find that while each branch individually yields comparable performance, the combination of both branches—enhanced by our graph-text alignment technique—leads to significant performance gains. This demonstrates the effectiveness of our proposed methods.

*D.2.4 Effect of Graph Encoder Choice.* We also investigate the impact of different graph encoders on the performance of GT2Vec, as shown in Table 6. We observe that the performance with GAT achieves the best results on both CommonsenseQA and OpenBookQA.

**Table 10: Impact of different graph node embedding initialization models on accuracy for the CommonsenseQA dataset.**

| Model | Output Dimension | Accuracy |
|---|---|---|
| RoBERTa-Large [38] | 1,024 | 81.09 ± 0.73 |
| E5-Small [59] | 384 | 80.37 ± 0.38 |
| E5-Base [59] | 768 | 80.77 ± 0.05 |
| E5-Large [59] | 1,024 | 80.79 ± 0.30 |
| E5-Mistral [60] | 4,096 | 80.42 ± 0.90 |

*D.2.5 Impact of Node Embedding Initialization.* To assess the impact of different node embedding initialization models on performance, we conducted experiments on the CommonsenseQA dataset. As shown in Table 10, the results reveal minimal differences in accuracy across models with varying output dimensions, indicating that the choice of initialization model has limited influence on performance. For instance, RoBERTa-Large, E5-Base, and E5-Large achieve nearly identical results, with an accuracy of around 81%.

Interestingly, even when using lighter models for node embedding initialization, such as E5-Small, which has a smaller output dimension (384), the performance remains competitive. This demonstrates that even when the node initialization model differs from the actual backbone used in GT2Vec, there is no significant performance degradation. For example, compared with E5-Mistral node embedding initialization, the lighter E5-Small model offers the advantage of reduced computational cost without compromising much on accuracy.