Harnessing the Power of Large Language Model for Uncertainty Aware Graph Processing

Zhenyu Qian[1†], Yiming Qian[2†], Yuting Song[2], Fei Gao[2], Hai Jin[1], Chen Yu[1], Xia Xie[3*]
[1]Huazhong University of Science and Technology, China
{qian_zhenyu, hjin, yuchen}@hust.edu.cn
[2]Institute of High Performance Computing (IHPC), Agency for
Science, Technology and Research (A*STAR), Singapore
{qian_yiming, song_yuting, gaofei}@ihpc.a-star.edu.sg
[3]Hainan University, China
shelicy@hainanu.edu.cn

Abstract

Handling graph data is one of the most difficult tasks. Traditional techniques, such as those based on geometry and matrix factorization, rely on assumptions about the data relations that become inadequate when handling large and complex graph data. On the other hand, deep learning approaches demonstrate promising results in handling large graph data, but they often fall short of providing interpretable explanations. To equip the graph processing with both high accuracy and explainability, we introduce a novel approach that harnesses the power of a large language model (LLM), enhanced by an uncertainty-aware module to provide a confidence score on the generated answer. We experiment with our approach on two graph processing tasks: few-shot knowledge graph completion and graph classification. Our results demonstrate that through parameter efficient fine-tuning, the LLM surpasses state-of-the-art algorithms by a substantial margin across ten diverse benchmark datasets. Moreover, to address the challenge of explainability, we propose an uncertainty estimation based on perturbation, along with a calibration scheme to quantify the confidence scores of the generated answers. Our confidence measure achieves an AUC of 0.8 or higher on seven out of the ten datasets in predicting the correctness of the answer generated by LLM.

Keywords: Graph Processing, LLM, Uncertainty, Knowledge Graph, Model Calibration

1. Introduction

Graph structure is one of the most common data structures in industry applications. It can store a large amount of information. Each node (vertex) in the graph represents an individual entity, point, or sample. Each edge (link) represents the relationships between the nodes in the graph. How to efficiently utilize the graph data on the downstream task is an active research area. Throughout time, researchers have proposed many approaches to make graph data processing more efficient and capable of handling larger graphs. There are three popular categories of graph processing techniques: matrix factorization methods, geometric methods, and deep learning-based algorithms. The matrix factorization methods formulate the graph structure data into adjacency matrices. The decomposition of such adjacency matrices can be used to indicate the potential links between nodes in the graph. This is developed based on the assumption that the relationship in the graph is mostly linear which can be represented by a low-rank matrix factorization (Nickel et al., 2011). Its simplicity enables the matrix factorization-based method ideal for processing small graphs but suffers scalability issues on large complex graphs. The geometric-based methods assume the relation between nodes or entities can be expressed as a linear combination (Bordes et al., 2013) or follows a certain topological pattern (Ebisu and Ichise, 2018). This assumption may not hold for graphs with complex data relations. To process large amounts of graph data with complex entity relations, deep learning-based methods, such as Graph Convolutional Networks (GCN) (Duvenaud et al., 2015), become widely used. It is a data-driven approach that does not rely on any assumptions about the entity relation structure. It enables graph processing algorithms to handle exponentially larger graphs. As the size of the model increases, the deep learning-based approach can handle complex data more accurately.

In this work, we follow the idea of leveraging large deep-learning models to handle graph data. Instead of using a widely recognized GCN or GNN model, we utilized pre-trained LLMs and then fine-tuned them with parameter efficient fine-tune (PEFT) which only 0.06% of the parameter was updated. We hypothesized that LLM pre-trained on billions of tokens already learned high-level abilities such as causal reasoning, learning with few examples, and language comprehension. We verified our hypothesis on two complex graph processing tasks: few-shot knowledge graph completion and graph classification.

† Authors contribute equally to this work
*Corresponding author

LREC-COLING 2024, pages 8035–8049
20-25 May, 2024.

Few-shot Knowledge Graph Completion is a task that enables the model to learn graph completion on new relations with a small set of data (1-10 samples). It helps with the problem of a large portion of knowledge graph relations being long-tail (Xiong et al., 2018). Furthermore, it enables applications that need to add new relations with a limited number of samples.

Graph Classification is a task that categorizes the graph into different classes. It has been widely used in medicine and science research where it is employed for tasks like analyzing small molecular properties analysis (Rodrigues et al., 2016), or large molecular protein classification (Borgwardt et al., 2005).

We conducted experiments using ten publicly available graph datasets in common knowledge, drug discovery, and organic chemistry. Our results demonstrate that fine-tuning a large language model with a simple LoRa (Hu et al., 2022) prompt tuning and a carefully designed prompt can greatly improve performance compared to current state-of-the-art algorithms.

Furthermore, we introduced an innovative uncertainty estimation method based on perturbation, along with a kernel density estimation (KDE) calibration approach to assess the model's confidence in its generated answers. Our results reveal that this confidence score can serve as a valuable feature for determining the correctness of the generated responses. Our algorithm achieved an impressive AUC of over 0.8 on seven of the ten datasets we investigated.

The contributions of our work can be summarized as follows:

1. We verified the scale-law holds for the few-shot learning setting on graph structure where the pre-trained LLMs with a higher number of parameters and pre-trained with larger datasets exhibit higher performance in graph processing tasks.

2. Our experiments conducted across diverse datasets covering various scientific domains indicate that a pre-trained LLM model with the LoRa prompt tuning and carefully designed prompt outperforms current state-of-the-art algorithms.

3. We propose a novel approach for uncertainty estimation based on perturbation, along with a model calibration scheme using Kernel Density Estimation (KDE) to quantify confidence scores.

2. Related Work

2.1. Graph Processing

Graph processing is a well-studied area. Researchers widely adopt three major categories of approaches: matrix factorization, geometric-based methods, and deep learning methods.

2.1.1. Matrix Factorization

The knowledge graph can be reformatted into an adjacency matrix, and decomposing this adjacency matrix into components can infer relationships between entities in incomplete graphs. Nickel et al., 2011 proposed a method called RESCAL, which is a three-way tensor χ with k slices. Each slice χ_k is factorized as $\chi_k \approx AR_kA^T$*, for k to , m.* r is the rank of the factorized matrix. A is $n \times r$ matrix that contains the latent-component representation of the entities corresponding to the row and column of the matrix χ_k. R_k is an asymmetric $r \times r$ matrix that models the interactions of the latent components in the k-th predicates. The score measures the relation k between the subject s and object o formulated as $\chi_{sok} = A_sR_kA_o^T$. Trouillon et al., 2016 suggested the relation presented in A can also be antisymmetric which decomposition only exists in the complex space. To handle this situation, the formulation is revised to $\chi_{sok} = Re(\langle R_k, A_s, \bar{A}_o \rangle)$, where Hermitian product is applied to handle the dot product with complex numbers. The real part of the Hermitian product is used as the relation score.

2.1.2. Geometric-based Methods

The other widely used approach for processing graph data is geometric-based methods. Bordes et al., 2013 proposed a method called TransE, which suggests that each entity within the graph can be expressed as a linear combination of its adjacent entity and relational attribute. However, this linear combination method encounters significant inaccuracies when entities engage in one-to-many $(1 - to - N)$ or many-to-many $(N - to - N)$ relationships with other entities. To tackle this challenge, Wang et al., 2014 proposed a method named TransH, which offers a solution by projecting the entity vector onto a hyperplane. This transformation effectively preserves the topological relationships between vectors. Both TransE and TransH assumed that entities and relations share the same vector space. However, this assumption may not always hold, as a single entity can possess multiple attributes. To overcome this limitation, Lin et al., 2015 proposed TransR, which eliminates this assumption by projecting the entity and relation vectors into different spaces. TransD (Ji et al., 2015) took a step further, offering a refined approach to the separation between entity and relation projections. Later, Ebisu and Ichise, 2018 proposed TorusE, which suggests that creating embedding in torus space brings the benefits of eliminating the need for regularization and avoiding embedding diverging into the same point.

2.1.3. Deep Learning-based Methods

As the availability of data increases, deep learning algorithms become the next optimal choice. The first natural choice is Graph Convolutional Network (GCN) (Duvenaud et al., 2015), which utilizes convolutional modules for processing graph structural data. Schlichtkrull et al., 2018 showed that the relational GCN structure is a well-fit candidate for tasks such as link completion and entity classification. Dettmers et al., 2018 suggested the work of heavy-duty GCN structure can be achieved by a lightweight convolutional neural network. Other network structures such as the capsule network-based model (Vu et al., 2019) and BERT (Yao et al., 2019) model achieved promising results as well. In recent years, LLMs have demonstrated strong capabilities in language generation, summarization, and other generative tasks. Leveraging the capabilities of LLMs (Yao et al., 2023) in knowledge graph completion has begun to demonstrate superior performance compared to previous methods in general large-scale knowledge graph completion tasks. Zhang et al., 2023a demonstrated that by using a prefix adapter, LLMs are capable of effectively discerning the correctness of triplets.

2.2. Model Uncertainty

Associations such as the European Commission have approved a law on AI system regulation (COMMISSION, 2021) which requires high levels of explainability in the AI model for many critical industry applications. In the field of explainability for large language models, the confidence or uncertainty score that LLM has on its generated answer is in focus. Four primary methods for generating uncertainty measures in LLMs include prompt design, answer consistency, token probability method, and supervised learning-based method.

Prompt design employs carefully designed prompts (Lin et al., 2022) to generate text together with its confidence score. Prompting strategies such as top-K (Tian et al., 2023) and chain-of-thought (Wei et al., 2022) can be utilized to further improve the performance.

Answer consistency measures the consistency of the generated answer under different LLM configurations, where higher consistency indicates the model has higher confidence in its answers. Methods such as different reasoning paths (Wang et al., 2023), self-generate variations (Ling et al., 2023), and perturbation of key tokens (Huang et al., 2023) enable LLMs to produce diverse answers for consistency measurement. Additionally, changing model configurations, such as adjusting temperature (Xiong et al., 2023) or enabling dropout during inference (Mo and Xin, 2023), can also generate answer variations.

Token probability method provides another way to measure the confidence of model outputs. The probability associated with each output token is leveraged to calculate uncertainties using methods such as mean token log-probability (Malinin and Gales, 2020), semantic entropy (Kuhn et al., 2023) and attention weighted entropy (Duan et al., 2023).

Supervised learning-based method utilizes human annotations to train a scoring model (Mielke et al., 2022) that predicts the probability of the model's response being correct.

3. Datasets

Our study focuses on evaluating the performance of LLMs on two tasks: few-shot knowledge graph completion (FKGC) and graph classification (GC), across 10 datasets.

Few-shot knowledge graph completion is evaluated on three datasets: NELL, Wiki and FB15K. The NELL and Wiki were proposed by Xiong et al., 2018, and we followed their instruction to split the dataset into training, validation, and testing three parts. The FB15K dataset is obtained from Toutanova and Chen, 2015, and the experimental setup is adapted from REFORM (Wang et al., 2021). The statistics of three datasets are summarized in Table 1.

Dataset	Ents.	Rels.	Triples	Few-Rels.
NELL	68,545	358	181,109	67
Wiki	4,838,244	822	5,859,240	183
FB15k	14,541	237	281,624	119

Table 1: Datasets statistics for few-shot knowledge graph completion task.

Graph classification on small graphs is evaluated on three molecular property prediction datasets: Tox21 (Huang et al., 2016), Sider (Kuhn et al., 2016) and ClinTox (Wu et al., 2018). The statistics of these three datasets are summarized in Table 2.

Graph classification on large graphs is evaluated on four datasets including two protein classification datasets: PROTEINS and ENZYMES obtained from Borgwardt et al., 2005, and two drug

Dataset	Tasks	Molecules
Tox21	12	7,831
Sider	27	1,427
ClinTox	2	1,478

Table 2: Dataset statistics for graph classification task on small graphs.

classification datasets: AIDS (Riesen and Bunke, 2008) and NCI1 (Wale et al., 2008). The statistics of these four datasets are summarized in Table 3.

Dataset	Graphs	Avg. |V|	Avg. |E|	Classes
PROTEINS	1,113	39.06	72.82	2
ENZYMES	600	32.63	62.14	6
AIDS	2,000	16.20	15.69	2
NCI1	4,110	32.30	29.87	2

Table 3: Dataset statistics for graph classification task on large graphs.

The train, validation, and test dataset splits for all datasets are summarized in Table 4.

Task	Dataset	# Train	# Val.	# Test
FKGC	Wiki	61,498	6,694	15,359
	NELL	8,526	1,004	2,213
	FB15k	14,221	3,056	8,320
GC	Tox21	65,005	7,208	7,360
	Sider	30,807	3,861	3,861
	ClinTox	2,364	296	296
	PROTEINS	889	112	112
	ENZYMES	480	60	60
	AIDS	1,440	360	200
	NCI1	2,960	739	411

Table 4: A summary of the dataset used in our experiments on few-shot knowledge graph completion (FKGC) and graph classification (GC) tasks. The number of train/validation/test samples are listed.

4. Method

The capabilities of pre-trained LLMs are widely recognized for effectively handling various tasks, such as sentiment analysis (Xu et al., 2019a), text summarization (Liu and Lapata, 2019), named entity recognition (Li et al., 2020b), sentence encoding (Reimers and Gurevych, 2019) et al. In this study, we investigate the feasibility of utilizing LLMs for processing knowledge graphs. We formulate the two knowledge graph tasks namely few-shot knowledge graph completion and graph classification as a classification problem. Moreover, we propose a novel uncertainty measure, which aims to assess the quality of the classification outcome.

4.1. Prompt Preparation

The models are fine-tuned through prompt tuning where we follow the template from Alpaca (Taori et al., 2023) to construct prompts. As shown in Table 5, the Alpaca template consists of four parts: task description, instruction, input, and response. We formulate the graph completion and graph classification tasks as multiple-choice questions. The task description section provides the information for the tasks. The instruction section provides the task requirements and available choices for the question. The input section contains the input samples. For example, in the graph completion task, the head and tail entities are used as input. While in the graph classification on the chemical molecular property analysis dataset, we use the target and SMILES string(Weininger, 1988) as input. The response section contains the index of the correct choice. We use the index instead of the actual answer as an easy way to help us catch the answer with incorrect formats. Additional examples of detailed prompts can be found in the Appendix.

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
Instruction: {instruction}
Input: {input}
Response:

Table 5: Alpaca Template for prompt tuning.

4.2. Model Selection

In 2017, the transformer (Vaswani et al., 2017) model surpassed models such as LSTM and RNN, becoming the primary backbone model for NLP tasks. Since then, many variances of transformer models have come out such as the unsupervised trained BERT (Devlin et al., 2019), and its improved version RoBerta (Liu et al., 2019). As NLP research progresses, exponentially larger language models such as the encoder-decoder structure T5 (Raffel et al., 2020) model and decoder-only structure OPT (Zhang et al., 2022) model are gaining traction. This indicates that the combination of a large amount of high-quality data and a large model size can achieve superior results. When it comes to text generation tasks, decoder-only structure models have become a more popular choice due to their capability to handle multiple tasks. Among the open-sourced language language models, the LLaMa (Touvron et al., 2023a) and the subsequent model LLaMa2 (Touvron et al., 2023b) are the most frequently used by researchers. It's worth mentioning that compared to LLaMa, the LLaMa2 model is trained with 40% more data and fine-tuned on over 1 million human preferences.

Our study focuses on investigating the feasibility of utilizing LLMs on two graph processing tasks. To select the best LLM, we test eight different representative LLMs on three datasets. The number of parameters for each selected model is 355M for RoBerta-L, 340M for BERT-L, 1.3B for OPT-1.3b, 6.7B for OPT-6.7b, 770M for T5-L, and 7B for Mistral-7b, LLaMa and LLaMa2. The performance of LLMs is summarized in Table 6. The ClinTox, ENZYMES, and NELL datasets are used to evaluate LLMs' performance on graph classification with small graphs, graph classification with large graphs, and few-shot knowledge graph completion tasks respectively. The results show LLaMa2-7b model achieves the highest average performance across tasks. Based on this observation, we choose the LLaMa2-7b model for all subsequent evaluations.

Models	ClinTox	ENZYMES	NELL
RoBerta-L	98.9%	81.7%	97.6%
BERT-L	97.6%	40.0%	90.1%
OPT-1.3b	99.5%	98.3%	91.8%
OPT-6.7b	99.4%	99.2%	95.4%
T5-L	99.2%	43.3%	29.1%
Mistral-7b	99.7%	98.3%	98.8%
LLaMa-7b	**99.8%**	93.3%	89.4%
LLaMa2-7b	99.6%	**99.2%**	**99.2%**

Table 6: The eight selected LLMs' performance on graph processing tasks. AUC is utilized as the metric for the ClinTox dataset, while hits@1 is employed as the metric for other datasets.

4.3. Model Fine-tuning

Due to the hardware resource limitation, we employ a parameter efficient fine-tuning method, LoRA (Hu et al., 2022), to fine-tune LLMs. The LoRA method freezes the weight of the pre-trained model and adds a small amount of additional weight to the existing model layers. For the selected LLaMa2 model, 0.06% (4.2M out of 7B) parameters are trained through the LoRa fine-tuning process.

5. Experiments

5.1. Implement Details

Our experiment is conducted on a server with four Tesla V100 32GB GPUs, Intel(R) Xeon(R) Gold 5117 CPU @2.00GHz, and 256GB RAM. FP16 is used during the experiment. The settings for LoRA fine-tuning are as follows: lr=0.0003, LoRA R=8, LoRA Alpha=16, LoRA Dropout=0.05, and the target modules for LoRA are q_proj and v_proj. [1]

5.2. Few-shot Knowledge Graph Completion

We follow the standard process as GMatching (Xiong et al., 2018) where the training comes in two steps. In the first step, we utilized the background data to train the system. Then the model is fine-tuned 5-shot learning where each type of relation has five samples. We evaluated three few-shot learning datasets of the graph completion task.

We compared our method with six baselines: Relation Network (Sung et al., 2018), Prototypical Network (Snell et al., 2017), GMatching (Xiong et al., 2018), FSRL (Zhang et al., 2020), FAAN (Sheng et al., 2020) and REFORM (Wang et al., 2021). As shown in Table 7, our LLaMa2-7b -based method achieves the best performance on all three datasets.

To conduct further analysis of the performance of LLMs in few-shot learning scenarios, we conducted experiments training the model with different background datasets. The term ”background dataset" refers to triplets that belong to the same field as subsequent training samples. Although they will not appear in later training, validation, or test sets, training the large language model on the background dataset first can provide the model with some basic knowledge of that field. We conducted experiments with 0-shot and 1-shot learning settings. The results are shown in Table 8. When the training and testing set are conducted on the different datasets, our performance

[1] The code is available at https://github.com/code4paper-2024/code4paper

Dataset	NELL	Wiki	FB15k
Relation Network	67.9	51.6	62.2
Prototypical Network	84.6	66.1	84.1
GMatching	90.8	75.2	84.8
FSRL	87.9	63.7	83.3
FAAN	94.7	74.3	86.3
REFORM	94.4	78.8	88.1
Ours	**99.2**	**83.1**	**89.9**

Table 7: Experimental results of 5-shot learning on knowledge graph completion task.(hits@1)

drops between 5 to 20%. It indicates training with background knowledge is important even for the LLaMa2-7b model that is pre-trained on 2 Trillion tokens. When the background dataset and the test dataset come from the same background, compared with 1-shot learning, the results of 0-shot learning immediately dropped around 20%. This indicates that the LLM can adapt to new graph links even with just one sample.

BG	Mode	Test	LLaMa2
NELL	1-shot	NELL	0.905
Wiki	1-shot	NELL	0.808
FB15k	1-shot	NELL	0.819
NELL	0-shot	NELL	0.616
NELL	1-shot	Wiki	0.614
Wiki	1-shot	Wiki	0.641
FB15k	1-shot	Wiki	0.597
Wiki	0-shot	Wiki	0.415
NELL	1-shot	FB15k	0.583
Wiki	1-shot	FB15k	0.806
FB15k	1-shot	FB15k	0.808
FB15k	0-shot	FB15k	0.580

Table 8: Few-shot learning graph completion accuracy(hits@1) with different background (BG) datasets.

5.3. Small Graph Classification

To evaluate the efficiency of LLM for graph classification on small graphs, we chose three chemical molecular property prediction datasets. The input of those datasets are strings in SMILES format (Weininger, 1988). Due to the small size of those datasets, a 10-fold nested cross-validation is utilized in the evaluation.

Our baselines include GraphLoG (Xu et al., 2021), AD-GCL (Suresh et al., 2021b), GraphCL (You et al., 2020), JOAO (You et al., 2021), SimGRACE (Xia et al., 2022a), GraphMAE (Hou et al., 2022), GraphMVP (Liu et al., 2022), MGSSL (Zhang et al., 2021), MoMu (Su et al., 2022), Mole-BERT (Xia et al., 2022b) and GIT-Mol (Liu et al., 2023). The evaluation results are shown in Table 9. Our LLM-based method outperformed all the baseline methods by a large margin.

Models	Tox21	Sider	ClinTox
GraphLoG	75.1	59.6	75.7
AD-GCL	74.9	61.5	77.2
JOAO	74.8	60.4	66.6
SimGRACE	74.4	60.2	75.5
GraphCL	75.1	59.8	77.5
GraphMAE	75.2	60.5	76.5
GraphMVP	74.9	60.2	79.1
MGSSL	75.2	61.6	77.1
MoMu	75.6	60.5	79.9
Mole-BERT	76.8	62.8	78.9
GIT-Mol	75.9	63.4	88.3
Ours	**79.6**	**82.1**	**99.6**

Table 9: Experimental results of graph classification on three molecular property prediction datasets. (AUC)

5.4. Large Graph Classification

We also evaluate the LLM's performance on large-scale graph classification. In the experiments, we use two protein-related datasets (ENZYMES, PROTEINS), and two drug classification datasets (NCI1, AIDS). For the protein-related datasets, since the amino acid sequences have long lengths and few types of amino acids, direct training on these datasets generated poor results. To solve this issue, we employ a technique similar to feature engineering, where consecutive identical amino acids are merged. For example, a sequence like [1, 1, 1, 2, 2, 3, 3, ...], where 1, 2, 3 represent different types of amino acids, is merged into [1(3), 2(2), 3(2), ...], with the numbers in parentheses indicating the consecutive occurrences.

We compared our methods with several baselines includes GCN (Kipf and Welling, 2017), GCNII (Chen et al., 2020), GIN (Xu et al., 2019b), HGP-SL (Zhang et al., 2023b), SAGE (Hamilton et al., 2017), DeeperGCN (Li et al., 2020a), GT (Dwivedi and Bresson), SAN (Kreuzer et al., 2021), SAT (Chen et al., 2022), GPS (Rampášek et al., 2022), UGT (Hoang et al., 2023), RW (Gärtner, 2003), FGW (Titouan et al., 2019), HSGE (Dutta et al., 2020), SGE (Dutta and Sahbi, 2018), CommPOOL (Tang et al., 2021), linearFGW (Nguyen and Tsuda, 2023), WL (Shervashidze et al., 2011) and XGraphBoost (Deng et al., 2021). As shown in Table 10, and 11, our method outperformed all the baselines on four datasets. As the datasets come from different domains, these results demonstrate the robustness of LLM to handle graph classification tasks across diverse domains.

Models	ENZYMES	PROTEINS
GCN	18.2	59.2
SAGE	21.5	62.8
GCNII	31.5	62.5
GIN	33.6	64.1
DeeperGCN	25.4	61.2
GT	41.7	77.3
SAN	22.5	68.5
SAT	50.9	62.9
GPS	62.7	53.8
UGT	67.2	80.1
HGP-SL	68.8	84.9
Ours	**99.2**	**99.6**

Table 10: Experimental results of graph classification on two protein datasets.(hits@1)

Models	NCI1	Models	AIDS
XGraphBoost	61.9	FGW	91.0
RW	69.0	RW	98.5
WL	82.5	HSGE	99.0
GCN	76.0	CommPOOL	98.5
SGE	82.5	SGE	98.7
GIN	79.1	linearFGW	98.7
Ours	**99.5**	Ours	**99.2**

Table 11: Experimental results of graph classification on NCI1 and AIDS datasets.(hits@1)

6. Model Uncertainty

In this work, we also propose to measure the model uncertainty from two perspectives: dataset level and sample level.

6.1. Dataset Level Uncertainty

Information entropy (Cover, 1999) is a well-known metric for quantifying the uncertainty of algorithm output. Notably, our observations reveal that the LLM exhibits a high degree of confidence in its output, with the cumulative probability of the top two selections accounting for approximately 99%. In our method, for each sample denoted as x, we extract the top two probabilities ($p_1(x)$ and $p_2(x)$) from the selection of outputs. Then, we compute the average information entropy across the test set, denoted as $H(\chi)$, for each sample.

$$H(\chi) = -\frac{1}{m}\sum_{x\in\chi}(p_1(x)\log p_1(x) + p_2(x)\log p_2(x)) \quad (1)$$

where $x \in \chi$, m denote the number of samples in the dataset χ. This process is iterated across all ten datasets. Figure 1 illustrates the relationship between entropy and the LLM's accuracy on different datasets. We observe a clear trend that datasets with higher information entropy tend to have lower accuracy.

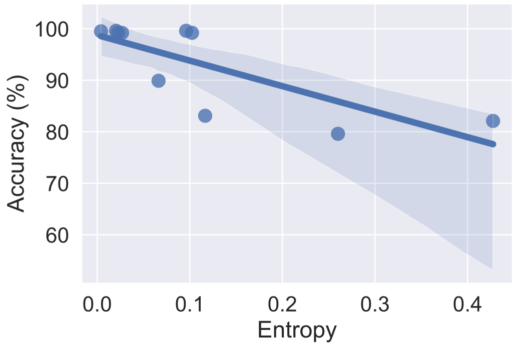

Figure 1: Evaluation of mean information entropy of the dataset vs. mean dataset accuracy.

6.2. Sample Level Uncertainty

6.2.1. Sample Uncertainty Extraction

We also proposed a novel approach for assessing sample-level uncertainty. This sample-level uncertainty can be used as a confidence score to indicate the correctness of the model's output. For a given sample denoted as x, we introduce a small random perturbation ϵ to create a new sample $x' = x + \epsilon$. The perturbation ϵ is chosen to be sufficiently small so that not have a material influence on the algorithm's output. Then, we use this new sample x' as an input for an LLM L to obtain the top-1 probability for the selection. This process is iterated k times, resulting in a list of probabilities denoted as $P = \{p_1, p_2, \dots p_k\}$. The sample's standard deviation, x, is then defined as follows:

$$\sigma_x = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(p_i - \bar{P})} \tag{2}$$

A challenge associated with using standard deviation as a measure of uncertainty is the absence of a reference point to determine whether it represents a substantial value. To address this issue, we suggest employing the validation set for calibrating the uncertainty metric.

6.2.2. Uncertainty Model Calibration

To illustrate the calculation process, we use the largest dataset Wiki-One as an example. The standard deviation σ_x of each example is collected from the validation set. A negative log is applied to each σ_x, and the histogram of the validation set for σ_x is shown in Fig. 2. From this histogram, we observe that the $-\log(\sigma)$ does not follow a normal distribution. To make our approach more flexible, we apply kernel density estimation (KDE) to estimate the probability density function (PDF). Let $(x_1, x_2, \ldots, x_n)$ be independent and identically distributed samples drawn from the validation dataset X_v where we set $n = 100$. The PDF of input variable x is estimated as follows:

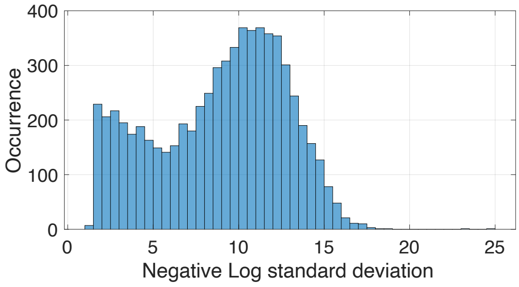

Figure 2: Histogram of negative log standard deviation for Wiki-One validation set.

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{3}$$

where K is the Gaussian kernel. The bandwidth is defined by Scott's rule (Scott, 1979):

$$h = \frac{\bar{X}_v}{0.6745}\left(\frac{4}{3m}\right)^{\frac{1}{5}} \tag{4}$$

where m is the number of points in X_v. We hypothesize that answers with higher standard deviation (lower negative log standard deviation) will have a higher probability of being a wrong answer to the query. Based on this hypothesis, we utilize the function $\hat{f}(x)$ to estimate cumulative density function (CDF) $\hat{F}(x)$ (shown in Fig. 3) which is formulated as follows:

$$\hat{F}(x) = \int_{x}^{-\infty} \hat{f}(t)dt \tag{5}$$

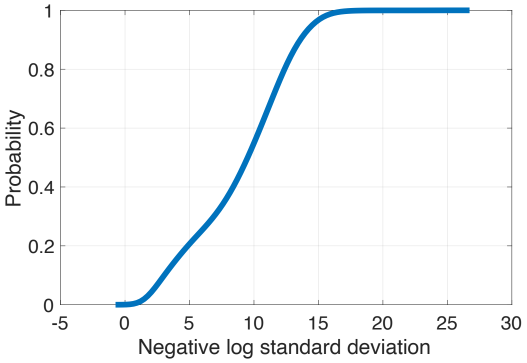

Figure 3: The cumulative density function estimated from the validation set of the Wiki-One dataset.

The CDF probability estimated indicates the confidence score for sample x. The distribution of confidence score from the test set is shown in Fig. 4. The positive label indicates the samples that received a correct answer from the LLM, while the negative label indicates the samples that received a wrong answer. This confirmed our hypothesis that the negative samples concentrate at lower confidence scores.

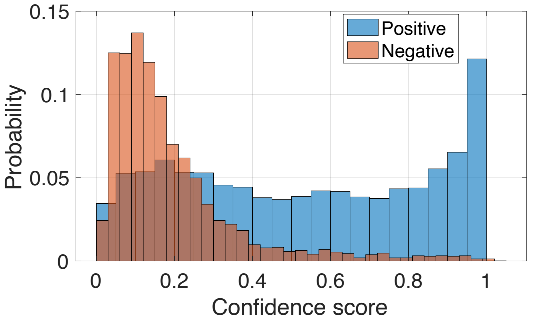

Figure 4: Distribution of confidence score for the positive (model gives a correct answer) and negative (model gives a wrong answer) from the Wiki-One test set.

We experimented with the confidence scores and ran a classification that categorizes samples as positive or negative labels. As shown in Fig. 5, our method received 0.82 AUC on the Wiki-One test set.

We expanded the experiments to all 10 datasets. The results are shown in Fig. 6. Our method achieved high AUC scores across various datasets, in which 7 out of 10 datasets achieved

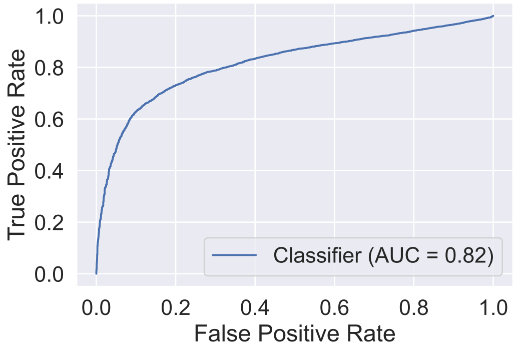

Figure 5: The ROC curve of using confidence score to classify the answer from samples from Wiki-One test set.

an AUC score higher than 0.8. This reveals our uncertainty measurement has the potential application to evaluate the quality of output without any additional changes to LLM.

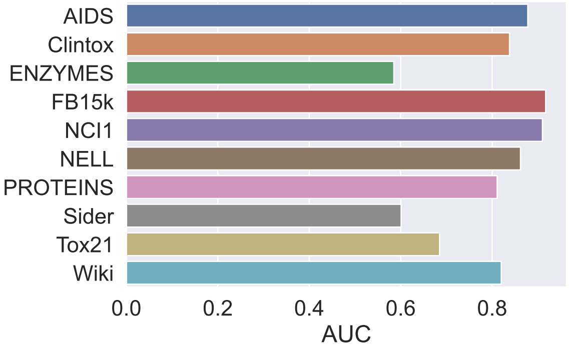

Figure 6: Results of applying our confidence score to classify positive and negative samples on 10 datasets.

7. Conclusion

In this paper, we introduced an approach that harnesses the power of a large language model for processing graph data. We conducted experiments across ten diverse datasets, on few-shot knowledge graph completion and graph classification tasks. Our method consistently outperformed state-of-the-art approaches across all ten datasets by a significant margin. Furthermore, we proposed a novel uncertainty-aware confidence scoring algorithm, which serves as an indicator of the correctness of answers generated by the large language model to enhance the model's explainability.

8. Acknowledgement

This paper has been supported by the National Natural Science Foundation of China (No. 62362023). We express our sincere gratitude for the financial assistance provided, which greatly facilitated the progress of our work.

9. References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. 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)*, pages 5185–5194, Hong Kong, China. Association for Computational Linguistics.

Camille Bilodeau, Wengong Jin, Tommi Jaakkola, Regina Barzilay, and Klavs F Jensen. 2022. Generative models for molecular discovery: Recent advances and challenges. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(5):e1608.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. 2005. Protein function prediction via graph kernels. *Bioinformatics*, 21:i47–i56.

Shaked Brody, Uri Alon, and Eran Yahav. 2022. How attentive are graph attention networks? In *ICLR*. OpenReview.net.

Dexiong Chen, Leslie O'Bray, and Karsten Borgwardt. 2022. Structure-aware transformer for graph representation learning. In *International Conference on Machine Learning*, pages 3469–3489. PMLR.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. 2020. Simple and deep graph convolutional networks. In *International conference on machine learning*, pages 1725–1735. PMLR.

Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta relational learning for few-shot link prediction in knowledge graphs. In *EMNLP/IJCNLP (1)*, pages 4216–4225. Association for Computational Linguistics.

Wenhu Chen, Wenhan Xiong, Xifeng Yan, and William Yang Wang. 2018. Variational knowledge graph reasoning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1823–1832.

EUROPEAN COMMISSION. 2021. Proposal for a regulation of the european parliament and of the council.

Thomas M Cover. 1999. *Elements of information theory*. John Wiley.

Daiguo Deng, Xiaowei Chen, Ruochi Zhang, Zengrong Lei, Xiaojian Wang, and Fengfeng Zhou. 2021. Xgraphboost: extracting graph neural network-based features for a better prediction of molecular properties. *Journal of chemical information and modeling*, 61(6):2697–2705.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Jinhao Duan, Hao Cheng, Shiqi Wang, Chenan Wang, Alex Zavalny, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2023. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *arXiv preprint arXiv:2307.01379*.

Anjan Dutta, Pau Riba, Josep Lladós, and Alicia Fornés. 2020. Hierarchical stochastic graphlet embedding for graph-based pattern recognition. *Neural Computing and Applications*, 32:11579–11596.

Anjan Dutta and Hichem Sahbi. 2018. Stochastic graphlet embedding. *IEEE Transactions on Neural Networks and Learning Systems*, 30(8):2369–2382.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28.

VP Dwivedi and X Bresson. A generalization of transformer networks to graphs. arxiv 2020. *arXiv preprint arXiv:2012.09699*.

Takuma Ebisu and Ryutaro Ichise. 2018. Toruse: Knowledge graph embedding on a lie group. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Thomas Gärtner. 2003. A survey of kernels for structured data. *ACM SIGKDD explorations newsletter*, 5(1):49–58.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Frank L Hitchcock. 1927. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189.

Van Thuy Hoang, O Lee, et al. 2023. Transitivity-preserving graph representation learning for bridging local connectivity and role-based similarity. *arXiv preprint arXiv:2308.09517*.

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

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 *ICLR*. OpenReview.net.

Ruili Huang, Menghang Xia, Dac-Trung Nguyen, Tongan Zhao, Srilatha Sakamuru, Jinghua Zhao, Sampada A Shahane, Anna Rossoshek, and Anton Simeonov. 2016. Tox21challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs. *Frontiers in Environmental Science*, 3:85.

Yuheng Huang, Jiayang Song, Zhijie Wang, Huaming Chen, and Lei Ma. 2023. Look before you leap: An exploratory study of uncertainty measurement for large language models. *arXiv preprint arXiv:2307.10236*.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, Beijing, China. Association for Computational Linguistics.

Xiaotian Jiang, Quan Wang, and Bin Wang. 2019. Adaptive convolution for multi-relational learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 978–987.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR (Poster)*. OpenReview.net.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. 2021. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Michael Kuhn, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. 2016. The sider database of drugs and side effects. *Nucleic acids research*, 44(D1):D1075–D1079.

Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. 2020a. Deepergcn: All you need to train deeper gcns. *arXiv preprint arXiv:2006.07739*.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020b. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Yuling Li, Kui Yu, Yuhong Zhang, and Xindong Wu. 2022. Learning relation-specific representations for few-shot knowledge graph completion. *arXiv preprint arXiv:2203.11639*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

Chen Ling, Xujiang Zhao, Xuchao Zhang, Yanchi Liu, Wei Cheng, Haoyu Wang, Zhengzhang Chen, Takao Osaki, Katsushi Matsuda, Haifeng Chen, et al. 2023. Improving open information extraction with large language models: A study on demonstration uncertainty. *arXiv preprint arXiv:2309.03433*.

Pengfei Liu, Yiming Ren, and Zhixiang Ren. 2023. Git-mol: A multi-modal large language model for molecular science with graph, image, and text. *arXiv preprint arXiv:2308.06911*.

Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. 2022. Pre-training molecular graph representation with 3d geometry. In *ICLR*. OpenReview.net.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1907.11692*.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing Conversational Agents' Overconfidence Through Linguistic Calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Shentong Mo and Miao Xin. 2023. Tree of uncertain thoughts reasoning for large language models. *arXiv preprint arXiv:2309.07694*.

Dai Hai Nguyen and Koji Tsuda. 2023. On a linear fused gromov-wasserstein distance for graph structured data. *Pattern Recognition*, 138:109351.

Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel, et al. 2011. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, pages 3104482–3104584.

Guanglin Niu, Yang Li, Chengguang Tang, Ruiying Geng, Jian Dai, Qiao Liu, Hao Wang, Jian Sun, Fei Huang, and Luo Si. 2021. Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion. In *Proceedings of the 44th International ACM SIGIR conference on research and development in information retrieval*, pages 213–222.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-gcn: Geometric graph convolutional networks. In *ICLR*. OpenReview.net.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. 2022. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515.

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*. Association for Computational Linguistics.

Kaspar Riesen and Horst Bunke. 2008. Iam graph database repository for graph based pattern recognition and machine learning. In *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, SSPR*, pages 287–297. Springer.

Tiago Rodrigues, Daniel Reker, Petra Schneider, and Gisbert Schneider. 2016. Counting on natural products for drug design. *Nature chemistry*, 8(6):531–541.

Michael 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 15*, pages 593–607. Springer.

David W Scott. 1979. On optimal and data-based histograms. *Biometrika*, 66(3):605–610.

Jiawei Sheng, Shu Guo, Zhenyu Chen, Juwei Yue, Lihong Wang, Tingwen Liu, and Hongbo Xu. 2020. Adaptive attentional network for few-shot knowledge graph completion. In *EMNLP (1)*, pages 1681–1691. Association for Computational Linguistics.

Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9).

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Bing Su, Dazhao Du, Zhao Yang, Yujie Zhou, Jiangmeng Li, Anyi Rao, Hao Sun, Zhiwu Lu, and Ji-Rong Wen. 2022. A molecular multi-modal foundation model associating molecule graphs with natural language. *arXiv preprint arXiv:2209.05481*.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208.

Susheel Suresh, Vinith Budde, Jennifer Neville, Pan Li, and Jianzhu Ma. 2021a. Breaking the limit of graph neural networks by improving the assortativity of graphs with local mixing patterns. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery*, pages 1541–1551.

Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. 2021b. Adversarial graph augmentation to improve graph contrastive learning. *Advances in Neural Information Processing Systems*, 34:15920–15933.

Haoteng Tang, Guixiang Ma, Lifang He, Heng Huang, and Liang Zhan. 2021. Commpool: An interpretable graph pooling framework for hierarchical graph representation learning. *Neural Networks*, 143:669–677.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *EMNLP*, pages 5433–5442. Association for Computational Linguistics.

Vayer Titouan, Nicolas Courty, Romain Tavenard, and Rémi Flamary. 2019. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, pages 6275–6284. PMLR.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, Beijing, China. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 2071–2080, New York, New York, USA. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. *stat*, 1050(20):10–48550.

Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, Dinh Phung, et al. 2019. A capsule network-based embedding model for knowledge graph completion and search personalization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2180–2189.

Nikil Wale, Ian A Watson, and George Karypis. 2008. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14:347–375.

Song Wang, Xiao Huang, Chen Chen, Liang Wu, and Jundong Li. 2021. Reform: Error-aware few-shot knowledge graph completion. In *Proceedings of the 30th ACM International Conference on Information*, pages 1979–1988.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1).

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

David Weininger. 1988. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. 2018. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530.

Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. 2022a. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM Web Conference 2022*, pages 1070–1079.

Jun Xia, Chengshuai Zhao, Bozhen Hu, Zhangyang Gao, Cheng Tan, Yue Liu, Siyuan Li, and Stan Z Li. 2022b. Mole-bert: Rethinking pre-training graph neural networks for molecules. In *The Eleventh International Conference on Learning Representations*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2018. One-shot relational learning for knowledge graphs. In *EMNLP*, pages 1980–1990. Association for Computational Linguistics.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019a. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019b. How powerful are graph neural networks? In *ICLR*. OpenReview.net.

Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. 2021. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*, pages 11548–11558. PMLR.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Liang Yao, Jiazhen Peng, Chengsheng Mao, and Yuan Luo. 2023. Exploring large language models for knowledge graph completion. *arXiv preprint arXiv:2308.13916*.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. 2021. Graph contrastive learning automated. In *International Conference on Machine Learning*, pages 12121–12132. PMLR.

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*, 33:5812–5823.

Hanwen Zha, Zhiyu Chen, and Xifeng Yan. 2022. Inductive relation prediction by bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5923–5931.

Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V. Chawla. 2020. Few-shot knowledge graph completion. In *Proceedings of AAAI Conference on Artificial Intelligence*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yichi Zhang, Zhuo Chen, Wen Zhang, and Huajun Chen. 2023a. Making large language models perform better in knowledge graph completion. *arXiv preprint arXiv:2310.06671*.

Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Chee-Kong Lee. 2021. Motif-based graph self-supervised learning for molecular property prediction. *Advances in Neural Information Processing Systems*, 34:15870–15882.

Zhen Zhang, Jiajun Bu, Martin Ester, Jianfeng Zhang, Zhao Li, Chengwei Yao, Huifen Dai, Zhi Yu, and Can Wang. 2023b. Hierarchical multi-view graph pooling with structure learning. *IEEE Trans. Knowl. Data Eng.*, 35(1):545–559.

A. Appendix: prompt examples

dataset	prompt example
Nell	Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ### Instruction: Select the most likely relationship between the two entities 1. automobilemakerdealersincountry 2. animal such as invertebrate 3... ### Input: sports league/mlb, coach/joe_torre ### Response: 17
Wiki	Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ### Instruction: Select the most likely relationship between the two entities 1.series spin-off 2.partnership with 3... ### Input: Royal Clipper, Valletta ### Response: 131
FB15K	Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ### Instruction: Select the most likely relationship between the two entities 1./government/political_party_tenure/politician 2./film/film_set_designer/film_sets_designed 3... ### Input: Austria, Vienna ### Response: 53
Tox21, ClinTox	Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ### Instruction: Classify if the given target will be activated by the molecular or not. Output 1 for activate, and 0 for not. ### Input: target:NR-AR, molecular:NC(=S)Nc1ccccc1 ### Response: 0
Sider	Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ### Instruction: Classify if the given side effect will be caused by the molecular or not. Output 1 for cause, and 0 for not. ### Input: side effect:Reproductive system and breast disorders, molecular:CN1CCCC1C2=CN=CC=C2 ### Response: 1
Enzymes	Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ### Instruction: Given a protein ID, sequence, and related protein IDs, classify the given protein as one of the enzyme types. Output a number between 0 and 5. ### Input: protein_id:131, sequence:1(5),2(9),3(4) ### Response: 4
Proteins	Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ### Instruction: Given a protein ID and its sequence, classify if the given protein is an enzyme or not. Output 1 for enzyme, and 0 for not. ### Input: protein_id:855, sequence:0(4),1(10) ### Response: 1
AIDS	Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ### Instruction: Given a compound ID and its sequence, classify if the given compound is AIDS antiviral or not. Output 1 for antiviral, and 0 for not. ### Input: compound_id:655, sequence:3(1),0(2),3(1),0(1),2(1),0(1),5(1) ### Response: 1
NCI1	Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ### Instruction: Given a compound ID and its sequence, classify if the given compound is positive or negative to cell lung cancer. Output 1 for positive, and 0 for negative. ### Input: compound_id:3858, sequence:1(6),2(5),3(12) ### Response: 1