Faithful Knowledge Graph Explanations for Commonsense Reasoning

Anonymous ACL submission

Abstract

While fusing language models and knowledge graphs has become common in commonsense question answering research, enabling faithful chain-of-thought explanations in these models remains an open problem. Our analysis reveals that one major weakness of current KG-based explanation methodologies lies in overlooking the faithfulness of path decoding during evaluation. This oversight leads to the distribution of the graph encoder often diverging from the original model predictions. To address this gap, we present two main contributions: (1) We propose and validate Text-GNN Fidelity in this specific context, to assess the reliability of the graph representation. (2) We introduce TeGDA (Text-Graph Distribution-aware Alignment), a novel algorithm that aligns the graph encoder with the target model to improve the faithfulness of subsequent explanations and that can be easily integrated into existing approaches. Our experiments and analysis show its potential to produce more faithful systems. Concretely, our work emphasises the neglected distributional misalignment problem in LM-KG reasoning models, which has been a latent source of spurious explanations.

1 Introduction

Question answering relies on explicit text and implicit domain knowledge (Hirschman and Gaizauskas, 2001). Pre-trained language models, fine-tuned for QA tasks, are essential in NLP (Khashabi et al., 2020), using extensive textual knowledge. For commonsense reasoning, knowledge graphs (KGs) like Freebase (Bollacker et al., 2008), Wikidata (Vrandečić and Krötzsch, 2014), and ConceptNet (Speer et al., 2017) are used, enhancing reasoning with their structured entity relationships (Ren et al., 2020; Ren and Leskovec, 2020; Ren et al., 2021). KGs also compensate for language models' (LM) limited factual memory (Li et al., 2022), providing insight into LM inference (Danilevsky et al., 2020).

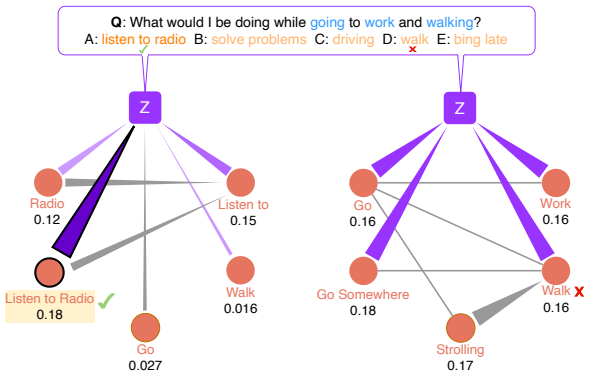

Figure 1: This figure depicts the attention weights assigned by two variants of the QA-GNN model when interpreting a reasoning path: the original implementation (right) and one trained using the TeGDA technique (left). A context node Z sits atop other concept nodes. In the original QA-GNN, near-equal attention weights around 0.17 are given to all nodes connected to Z. By comparison, the TeGDA approach resolves this limitation by assigning the highest attention weight to the most likely answer node, and markedly ten times smaller weights to other unrelated concepts. This differential weighting enhances model explainability by highlighting the pivotal connections influencing predictions.

Effective explanations should accurately reflect a model's reasoning (Herman, 2017). In knowledge-augmented commonsense QA, attention weights from message-passing provide explanations (Lin et al., 2019; Yasunaga et al., 2021), as in Figure 1. However, the reliability of these explanations is questionable (Jain and Wallace, 2019), and criteria for evaluating model explainability are often neglected, diminishing their impact.

We argue that explanations from a broad class of KG-enhanced LMs (LM-KG) are of limited faithfulness. The behaviour of graph encoder deviates from the overall LM-KG model and it has limited influence on the prediction, so explanations extracted from the graph encoder are unlikely to reflect the full set of facts. Besides, this process

does not guarantee that the extracted explanations will be faithful to the reasoning of the model (Jain and Wallace, 2019).

To advance faithful KG explanations, we introduce a novel metric, **Text-GNN Fidelity**, for assessing graph encoder faithfulness in LM-KG models. Prior works have evaluated GNN explanation faithfulness (Zhao et al., 2023), but ours is the first to propose quantitative fidelity metrics for KG explanations in LM-KG fusion models. Our findings reveal low fidelity in existing LM-KG models with a notable divergence between graph encoder and full model distributions. To address this, we propose **Text-GNN Distribution-aware Alignment** (TeGDA), featuring consistency regularisation and a method for extracting post-hoc explanations from the optimized GNN encoder. Our analysis, using CommonsenseQA and OpenBookQA datasets, shows that TeGDA enhances fidelity in various LM-KG models, marking a significant contribution to graph explainability and setting a benchmark for future research.

Research questions. This study aims to critically examine the extent to which graph explanations align with the reasoning processes of fusion models, thereby offering a metric for assessing model consistency. The research is structured around refined questions, focusing on the fidelity and impact of graph-based explanations: our work delves into the following:

Q1 How can we define and measure faithfulness in the context of discrepancies between graph encoder outputs and black-box language model predictions?

Q2 How prevalent is the issue of inconsistency in current graph explanation methods, and how to enhance the explainability of GNN encoders in terms of fidelity?

Q3 To what extent can TeGDA robustly identify explanatory sub-structures for interpreting the target graph model?

2 Related Work

Knowledge Graphs in NLP. Research has explored enhancing NLP with additional knowledge. Studies have shown pre-trained language models can serve as implicit knowledge bases (Pan et al., 2019; Petroni et al., 2019). Others have integrated structured knowledge graphs into language models for better knowledge representation, focusing on processing the knowledge graph (KG) and the language model (LM) separately before combining them for question answering (QA) tasks (Mihaylov and Frank, 2018; Wang et al., 2019; Zhang et al., 2022; Lin et al., 2019; Yasunaga et al., 2021).

Multi-relational Graph Encoder. Graph Neural Networks (GNNs) are significant in handling diverse graph structures (Kipf and Welling, 2017; Veličković et al., 2018). For multi-relational graphs like KGs, which have complex relational data, R-GCNs and GAT have been developed to handle these relations effectively (Schlichtkrull et al., 2018; Veličković et al., 2018).

KGs for Post-hoc Explanations in LMs. LMs struggle with interpretability (Danilevsky et al., 2020). Grounding LM outputs in KGs has been a method to provide explanations, but these are often not fully representative due to the reliance on text and graph embeddings (Feng et al., 2020; Sun et al., 2022; Jain and Wallace, 2019; Zhang et al., 2022; Yasunaga et al., 2021). Recent approaches like GraphMask attempt to improve faithfulness in explanations, but challenges persist in quantifying the fidelity of graph encoder explanations in LM-KG models (Schlichtkrull et al., 2021; Aglionby and Teufel, 2022).

3 Background

3.1 Models

In this study, we focus on a category of models that synergize a text encoder (LM) and a graph encoder for the purpose of commonsense question answering. These models effectively combine linguistic and structured world knowledge to enhance reasoning and understanding. In a multi-choice commonsense question answering setting, the model processes a question q and a set of answer choices $\mathcal{C}$. For each answer choice $a \in \mathcal{C}$, a concatenated input statement $\boldsymbol{s} = [\boldsymbol{q}; \boldsymbol{a}]$ is formed, where $\boldsymbol{q}$ and $\boldsymbol{a}$ represent the word entities in the question and answer choice, respectively. The external Knowledge Graph is then utilized to extract a relevant subgraph $\mathcal{G}$, guided by the input statement $\boldsymbol{s}$. This contextualized subgraph is formally defined as a multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{I}, \phi)$, where $\mathcal{V}$ represents the set of vertices (or nodes), $\mathcal{I}$ the set of edges, and ϕ the relational types in the graph. The language model, denoted as $\mathcal{LM}$, computes the context tokens $\mathcal{Z}_{\text{LM}} = \mathcal{LM}(s)$. This involves

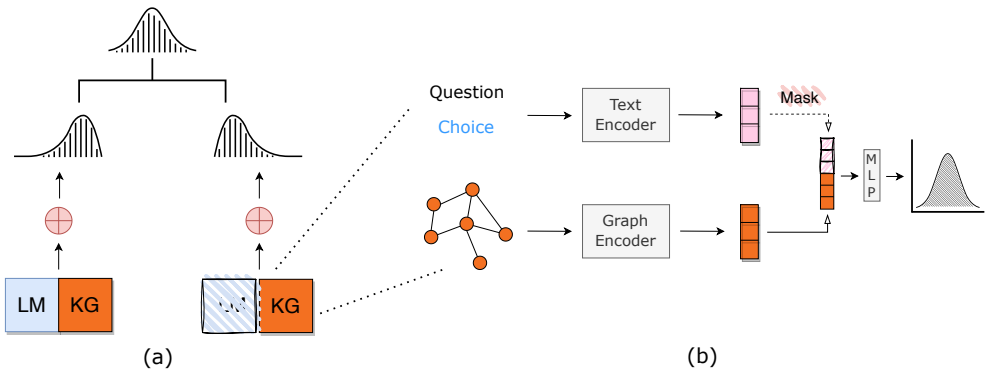

Figure 2: As depicted in (a), TeGDA comprises two primary components: the original model (left) and the LM-masked ablation (right). Within each model, LM and KG make interaction with each other. Following the fusion, predictions are generated independently. The losses from both are then simultaneously back-propagated to ensure greater consistency in the output distributions. (b) This part of the figure provides an in-depth representation of a representative LK-KG question answering model architecture discussed here. It also elucidates the details of the masking process and emphasises a category of question answering models that harness the capabilities of knowledge graphs.

encoding the concatenated question and answer choice into a high-dimensional vector space, capturing the linguistic nuances and semantic relationships. Simultaneously, a graph encoder $\mathcal{E}_{\text{KG}}$ is employed to encode the KG subgraph $\mathcal{G}$. The encoding $\mathcal{E}_{\text{KG}}(\mathcal{G})$ captures the structured relational information and knowledge present in the graph. Finally, a fusion module $\mathcal{F}$ integrates the outputs of both the $\mathcal{LM}$ and $\mathcal{E}_{\text{KG}}$ encoders to generate a joint representation $\mathcal{F}(\mathcal{Z}_{\text{LM}}, \mathcal{E}_{\text{KG}})$. This module can range from simple feature concatenation to more complex architectures, such as a transformer-based fusion layer, which effectively merges the linguistic context with the structured knowledge graph information. The output of this fusion model is then utilized to predict the correct answer from the set of choices, leveraging both the unstructured text understanding from the $\mathcal{LM}$ and the structured commonsense knowledge from the $\mathcal{G}$.

3.2 Graph Neural Network Architecture

Following Zhang et al. (2022), the graph encoder, denoted as $\mathcal{E}_G$, processes a local knowledge graph G linked to a question-answer (QA) example. Initially, it assigns initial embeddings $\{v_1^{(0)}, \ldots, v_J^{(0)}\}$ to the graph's nodes using pre-trained embeddings. In each Graph Neural Network (GNN) layer, these embeddings $\{v_0^{(\ell-1)}, v_1^{(\ell-1)}, \ldots, v_J^{(\ell-1)}\}$ are updated through information exchange among nodes, leading to updated node embeddings for each entity. Here, v_0 typically represents the context node:

$$\{v_0'^{(\ell)}, \ldots, v_J'^{(\ell)}\} = \mathcal{E}_G(\{v_0^{(\ell-1)}, \ldots, v_J^{(\ell-1)}\}) \quad \text{for } \ell = 1, \ldots, M \tag{1}$$

This process uses a modified graph attention network (GAT), similar to Yasunaga et al. (2021). The GNN calculates node representations $v_j'^{(\ell)}$ for each node v_j through message passing:

$$v_j'^{(\ell)} = f_n\left(\sum_{v_s \in \mathcal{N}_{v_j} \cup \{v_j\}} \beta_{sj} m_{sj}\right) + v_j^{(\ell-1)} \tag{2}$$

Here, $\mathcal{N}_{v_j}$ is the neighbourhood of node v_j, m_{sj} is the message from neighbour v_s to v_j, β_{sj} is an attention weight, and f_n is a two-layer Multilayer Perceptron (MLP). The message m_{sj} is calculated as:

$$r_{sj} = f_r(t_{sj}, u_s, u_j) \tag{3}$$

$$m_{sj} = f_m(v_s^{(\ell-1)}, u_s, r_{sj}) \tag{4}$$

Here, u_s, u_j are node type embeddings, t_{sj} is a relation embedding, f_r is a two-layer MLP, and f_m is a linear transformation. The attention weights β_{sj} are calculated as follows:

$$\beta_{sj} \propto \text{softmax}\left(f_q\left(v_s^{(\ell-1)}, u_s\right)^\top \cdot f_k\left(v_j^{(\ell-1)}, u_j, r_{sj}\right)\right) \tag{5}$$

Here, β_{sj} represents the attention weight between nodes s and j. The function f_q computes a query vector for the source node v_s, and f_k computes a key vector for the target node v_j. The dot product

of these vectors determines the raw attention score, which is then normalized across all neighbouring nodes using the softmax function. This simplified equation captures the core of the attention mechanism without the intricate details.

3.3 Post-hoc Explanation Extraction

Post-hoc explanations can be extracted from the trained fusion model by inspecting the attention weights of the final GNN layer. As the model performs multiple rounds of message passing and updating node representations, the attention weights in the last GNN layer indicate the most salient relationships in the graph with respect to predicting the final answer.

Specifically, for each attention head, we compute the average attention weight $\bar{\beta}_{ij}$ between all node pairs (i, j) connected by an edge in the graph. By averaging across multiple attention heads, we derive explanations that summarize the most important semantics captured in the full graph structure. More formally, let $\beta_{ij}^{h,M}$ indicate the attention weight between nodes i and j for the h-th attention head in the M-th (final) GNN layer. Then the explanation E_{ij} for edge (i, j) is:

$$E_{ij} = \frac{1}{H} \sum_{h=1}^{H} \beta_{ij}^{h,M} \tag{6}$$

where H is the total number of attention heads. By selecting edges from the last GNN layer that influence the joint prediction, these edge weights indicate the semantic paths in the graph that play a pivotal role connecting the question to the right answer choice. The highlighted subgraphs help to explain the reasoning process, as shown in figure 1.

4 Datasets

We assess our methods by using two multiple-choice question answering datasets: CommonsenseQA (Talmor et al., 2019) and OpenBookQA (Mihaylov et al., 2018), serving as benchmarks for commonsense reasoning.

CommonsenseQA. A dataset of 12,102 questions in a 5-way multiple-choice format which requires commonsense knowledge beyond mere language understanding. For our experiments, we adopted the **in-house (IH)** data split by Lin et al. (2019) to facilitate comparison with established baseline methods.

OpenBookQA. A dataset with its 4-way multiple-choice structure, assesses elementary scientific knowledge through its collection of 5,957 questions, accompanied by a compilation of scientific facts. For this dataset, we relied on the official data splits provided by Mihaylov et al. (2018).

5 Evaluating KG Faithfulness

We argue that LM-KG models are intrinsically unable to provide graph-structured explanations that are highly faithful to the full model. Our desire for these explanations is that they are the collection of facts used by the model to complete a natural language understanding task. The more faithful these explanations are, the more useful they will be for developers and users to understand model behaviour.

Assumption 1. If a **faithful explanation** is to be extracted from a GNN encoder, the output distributions of it should exhibit consistency or less discrepancy from the output of the original LM-KG model.

The measurement of the faithfulness of an explanation refers to how accurately it reflects the true reasoning process of the model (Herman, 2017). To answer **Q1**, here we propose the metrics for measuring the GNN explainability faithfulness in the context of LM-KG systems: **Text-GNN Fidelity**. In our case, if assumption 1 holds, the explanations generated by the graph encoder can serve as an interpretable proxy for the overall predictions made by the model.

5.1 Text-GNN Fidelity

Text-GNN Fidelity ($\boldsymbol{F_{\mathbf{TG}}}$) is defined as the overlap between the original model prediction and text-encoder-masked Multilayer Perceptron (MLP) predictions. Different from the methods by Aglionby and Teufel (2022), which measure the accuracy of newly trained ablation with text encoder output detached from the model or with text encoder frozen all the time. To maintain the integrity of the model, we directly mask out text encoder output from the fusion layer without new training and model modification (Figure 2b).

5.1.1 Probing by Masking

Inspired by (Schlichtkrull et al., 2021), $\boldsymbol{F_{\mathbf{TG}}}$ is conducted using a controlled variable method with masking, all factors are kept constant except that

the text encoder outputs are masked out. Keeping all parameters and the model architecture as is allows us to establish a causal relationship between the text encoder variable and the observed outcomes, especially in such a model class with multiple deep fusion layers. Masking here can be equivalently thought of as adding a certain type of noise when prediction, it contains at best minimal useful information for answering the question correctly. It can be categorised as belonging to the class of perturbation-based methods (Guan et al., 2019; Schlichtkrull et al., 2021).

Specifically, we calculate the F_{TG} as the statistical difference of outputs by checking the prediction overlap between models. **Text-GNN Fidelity** is defined as follows:

$$\begin{aligned} \boldsymbol{F}_{\mathbf{TG}} &= \frac{\sum \delta\left(c_{\mathcal{M}}, c_{\mathcal{M}_{\text{mask}}}\right)}{N} \\ \boldsymbol{c}_{\mathcal{M}} &= \arg\max_{c \in C} P(\hat{Y} = c \mid \mathcal{G}, \mathcal{M}) \\ \boldsymbol{c}_{\mathcal{M}_{\text{mask}}} &= \arg\max_{c \in C} P\left(\hat{Y} = c \mid \mathcal{G}, \mathcal{M}_{\text{mask}}\right) \end{aligned} \tag{7}$$

The $\boldsymbol{F}_{\mathbf{TG}}$ score, as a percentage of overlap, provides a measure of the agreement between the original model's output and the text-encoder-masked model's output. Where C is the set of choices, $c_{\mathcal{M}}$ is the prediction using the original model $\mathcal{M}$ and $\mathcal{M}_{\text{mask}}$ is the $\mathcal{Z}_{\text{LM}}$ masked model. $P(Y \mid \mathcal{M})$ denotes the probability distribution of the output Y given the model $\mathcal{M}$. $\delta(x, y)$ is the Kronecker delta function, which is 1 if $x = y$ and 0 otherwise, N is the total number of instances in the dataset considered. This represents the proportion of instances where the predictions from the two models agree. Measurement of $\boldsymbol{F}_{\mathbf{TG}}$ is reported in Table 2. We report the original models and the LM-disabled model performance in Table 1.

6 Improving Text-GNN Fidelity

To achieve a more faithful GNN interpretation, it's imperative to ensure that the introduced modifications of LM-KG models do not substantially deviate from the KG's behaviour, implying that even after introducing modifications, the GNN encoder should output a distribution that mirrors the one emitted by the unaltered model. While traditional methods have relied heavily on cross-entropy as the primary loss function, the inconsistent GNN encoder of existing LM-KG models demands a more nuanced approach. To answer **Q2** We next introduce **Text-GNN Distribution-aware Alignment** (TeGDA) – a strategy designed to bridge this gap (Figure 2).

6.1 Text-GNN Distribution-aware Alignment (TeGDA)

In order to quantitatively assess the divergence between the output density of our original model $\mathcal{M}$ and its masked variant $\mathcal{M}_{mask}$, we first devise the **Text-GNN Consistency** ($\boldsymbol{C}_{\mathbf{TG}}$) metric to measure the alignment between the probability distributions of their outputs. Our chosen metric is inspired by the Jensen–Shannon divergence $\mathcal{J}$ (Lin, 1991), a symmetrised and smoothed version of the Kullback-Leibler divergence (Kullback and Leibler, 1951), which offers a bounded measure of similarity between probability distribution pairs. The $\boldsymbol{C}_{\mathbf{TG}}$ metric is computed as follows:

$$\begin{aligned} \boldsymbol{C}_{\mathbf{TG}} : \mathcal{J}\left(\mathcal{M}_{\text{mask}}, \mathcal{M}\right) = \lambda \boldsymbol{D}_{\text{KL}}\left(P\left(Y \mid \mathcal{M}_{\text{mask}}\right) \| \mathcal{A}\right) \\ +(1-\lambda) \boldsymbol{D}_{\text{KL}}(P(Y \mid \mathcal{M}) \| \mathcal{A}) \end{aligned} \tag{8}$$

Where $\boldsymbol{D}_{\text{KL}}$ represents the Kullback-Leibler divergence. The key to the computation of $\mathcal{J}$ is the average of the two distributions, which we denote as $\mathcal{A}$:

$$\mathcal{N} = \frac{1}{2}\left(P\left(Y \mid \mathcal{M}_{\text{mask}}\right) + P(Y \mid \mathcal{M})\right) \tag{9}$$

$\mathcal{A}$ serves as the mid-point reference distribution against which the divergence of each of the two distributions is measured. By employing $\mathcal{J}$ as our metric for $\boldsymbol{C}_{\mathbf{TG}}$, we aim to capture the nuanced differences between the output probability distributions of $\mathcal{M}$ and $\mathcal{M}_{\text{mask}}$. A smaller $\mathcal{J}$ indicates a high degree of similarity or consistency between the two models, while a larger value signifies a greater divergence in their outputs. This assessment is crucial for understanding the impact of the GNN part model's behaviour and ensuring that the explainability remains intact across variants. A smaller $\boldsymbol{C}_{\mathbf{TG}}$ indicates that, even when the LM output is masked out, the graph encoder can still assign probabilities to choices that closely align with the original model's decisions, making it potentially more representative of the original model's thought process.

TeGDA enhances the cross-entropy $\mathcal{L}_{\text{CE}}$ by introducing a consistency factor $\mathcal{L}_{C_{\text{TG}}}$. This factor is a component that ensures the graph encoder's outputs align closely with the original model's predictions. The objective function for TeGDA is given by:

$$\mathcal{L}_{\text{align}}(\mathcal{M}, \mathcal{M}_{\text{mask}}) = \epsilon_1 \cdot \nabla_{\theta_t} \mathcal{L}_{\text{CE}} + \epsilon_2 \cdot \nabla_{\theta_t} \mathcal{L}_{C_{\text{TG}}} \quad (10)$$

In this equation, θ_t are the model parameters at time step t, ϵ_1, ϵ_2, are learning rates, $\mathcal{L}_{\text{CE}}$ represents the cross-entropy loss, which was traditionally employed. $\mathcal{L}_{C_{\text{TG}}}$ is the consistency term that measures the divergence between the probability distributions of the original and masked models. The equation shows the parameter update rule, where the gradients of the two losses are subtracted from the current parameters θ_t to obtain the updated parameters θ_{t+1}. Next, we go into the details of this strategy which also incorporates details on post-hoc explanation extraction:

- TeGDA utilizes text $\boldsymbol{s} = [\boldsymbol{q}; \boldsymbol{a}]$ and a background subgraph $\mathcal{G}$ for training. It connects entities from question q and answer $a \in \mathcal{C}$ in $\mathcal{G}$.The model first uses a language model encoder for text representations $\mathcal{Z}_{\text{LM}}$ and a graph neural network (GNN) encoder for graph embeddings $\mathcal{E}_{\text{KG}}$ of $\mathcal{G}$.

- A fusion module $\mathcal{F}$ then combines $\mathcal{Z}_{\text{LM}}$ and $\mathcal{E}_{\text{KG}}$ for answer prediction. Masking $\mathcal{Z}_{\text{LM}}$ generates $\mathcal{M}_{\text{mask}}$, used to calculate prediction probability $P(Y|\mathcal{M}_{\text{mask}})$.

- TeGDA minimizes the Jensen-Shannon divergence $\mathcal{J}$ between $P(Y|\mathcal{M})$ and $P(Y|\mathcal{M}_{\text{mask}})$ for better graph explanation and reasoning fidelity, using a joint objective $\mathcal{L}$ that includes both $\mathcal{J}$-based and cross-entropy terms.

- Post-hoc explanations are derived from the trained graph encoder $\mathcal{E}_{\text{KG}}$ by analyzing attention weights $\beta_{ij}^{h,M}$, indicating key semantic relationships in $\mathcal{G}$. These post-hoc explanations, aligned with the graph encoder's training, offer a more faithful reflection of the reasoning process of model $\mathcal{M}$.

7 Experiment Settings

7.1 Knowledge Graph

We use **ConceptNet** (Speer et al., 2017), a broad knowledge graph, for our tasks. A subgraph $\mathcal{G}$ for each QA context is extracted using the method by Feng et al. (2020) with hop size k=2.

7.2 Implementation & Training Details

Our model, following Feng et al. (2020); Yasunaga et al. (2021), includes a 4-head, 5-layer graph encoder (dimension $D = 200$) with a 0.2 dropout rate (Srivastava et al., 2014). Using RAdam (Liu et al., 2019a) with batch size 128, we refine parameters. Input node features from concatenated $[\boldsymbol{q}; \boldsymbol{a}]$ pass through RoBERTa-Large, yielding $1024d$ token embeddings. Gradient clipping at 1.0 (Pascanu et al., 2013) and learning rates of $1e^{-5}$ (LM) and $1e^{-3}$ (GNN) are set. Training takes about 3 hours for 30 epochs on a Tesla V100 GPU, with hyperparameters tuned on the development set.

7.3 LM-KG Baseline Models

To assess our TeGDA training and Text-GNN Fidelity metric, we compare it with three LM-KG models: QA-GNN(Yasunaga et al., 2021), GreaseLM(Zhang et al., 2022), and MHGRN(Feng et al., 2020), each contributing uniquely to integrating language models with knowledge graphs. QA-GNN introduces a context node for joint reasoning. GreaseLM enhances the interaction between language models and knowledge graphs through a fusion mechanism. MHGRN offers a graph encoding architecture for multi-hop relational reasoning over knowledge graphs.

For fair comparison, we use RoBERTa-Large (Liu et al., 2019b) in all the baselines and our methods. Under our new approach and node embedding settings, the performance of each behaves differently than described in papers, which we discuss next.

8 Results Analysis

Table 1 presents results on CommonsenseQA and OpenBookQA using TeGDA-trained models and three LM-masked models. TeGDA notably enhances faithfulness across all scenarios, with GreaseLMMask on the CommonsenseQA **IH**-dev split achieving a 36.2% accuracy increase. This highlights TeGDA's effectiveness in addressing model inconsistencies and bolstering graph encoder predictions, setting a foundation for reliable graph interpretation. Additionally, Table 3 reports $\boldsymbol{F}_{\mathbf{TG}}$ metric scores under TeGDA, showing consistent improvements, like over 98% faithfulness for GreaseLM across scenarios.

Method	CommonsenseQA		OpenBookQA	
	IH-dev (%)	**IH**-test (%)	Dev (%)	Test (%)
QA-GNN (Yasunaga et al., 2021)	76.5	**75.4**	78.2	80.8
+Mask	67.8	65.6	39.2	43.2
QA-GNN$_{\text{TeGDA}}$	75.8	69.8	79.0	80.0
+Mask	75.4 (↑ 7.6%)	69.8 (↑ 4.2%)	78.2 (↑ **39.0**%)	80.4 (↑ **37.2**%)
GreaseLM (Zhang et al., 2022)	**77.8**	73.0	**82.2**	**82.6**
+Mask	39.4	38.7	54.2	56.4
GreaseLM$_{\text{TeGDA}}$	76.2	71.1	80.6	82.4
+Mask	75.6 (↑ **36.2**%)	70.8 (↑ **32.1**%)	80.4 (↑ 26.2%)	**82.6** (↑ 26.2%)
MHGRN (Feng et al., 2020)	**77.8**	74.1	69.4	67.4
+Mask	48.4	46.6	60.6	56.6
MHGRN$_{\text{TeGDA}}$	76.9	71.2	71.2	66.6
+Mask	75.3 (↑ 26.9%)	68.8 (↑ 22.2%)	69.2 (↑ 8.6%)	66.6 (↑ 10.0%)

Table 1: Accuracy comparison of three different LM-KG models in their original version and trained with the TeGDA scheme (grey background) across two benchmark datasets. +Mask means the hidden state from the text encoder or from the interaction node of the model is masked out.

Original	CommonsenseQA		OpenBookQA	
	dev	test	dev	test
QA-GNN	**78.3**	**75.5**	39.3	45.5
GreaseLM	41.2	40.7	60.3	62.7
MHGRN	52.3	51.0	**75.4**	**73.0**

Table 2: Text-GNN Fidelity of the original model on each dataset.

TeGDA	CommonsenseQA		OpenBookQA	
	dev	test	dev	test
QA-GNN	98.5	**98.7**	97.6	98.0
GreaseLM	**98.9**	98.0	**99.6**	**99.6**
MHGRN	95.5	95.0	96.2	97.4

Table 3: Text-GNN Fidelity of models trained with the TeGDA scheme.

8.1 LM-masked Models

Performance results in Table 1 show that masking the text encoder leads to a significant drop in performance across all models on both datasets. For example, on CommonsenseQA **IH**-dev, masking reduces accuracy by ↓ 39.7% for GreaseLM. This substantial decrease demonstrates the text encoder's contribution to the overall reasoning and prediction process. In contrast, Table **??** shows the accuracy of **TeGDA**, the LM-disabled models (e.g. MHGRN$_{\text{Mask}}$ with a ↓ 2.4% and GreaseLM$_{\text{Mask}}$ with only ↓ 0.3% on CommonsenseQA **IH**-test split) exhibit only minor drops or even slight improvements (e.g. GreaseLM$_{\text{Mask}}$ with ↑ 0.2% on OpenBookQA **Test** split) in accuracy compared to the unmasked originals.

However, a tradeoff is that the unmasked models trained with TeGDA tend to have slightly lower accuracy than their counterparts without consistency regularisation. For example, on the OpenBookQA test set, MHGRN$_{\text{TeGDA}}$ accuracy is 66.6% compared to 67.4% for vanilla MHGRN. This decrease suggests enforcing consistency introduces some limitations on the modelling flexibility. Additionally, for QA-GNN, there is a small gap (↓ 8.7%) between the original model and masked model accuracy in Table 1. This indicates the GNN encoder outputs alone can achieve comparable performance to the full model, which is preferred. For GreaseLM and MHGRN, however, it shows that the text encoder makes a more significant contribution to these models.

In summary, the results show that TeGDA training improves graph encoder fidelity and reduces reliance on the text encoder. This supports the claim that TeGDA produces more faithful graph-based predictions that better reflect the full model's reasoning process. The graph encoder trained with TeGDA can then serve as a more reliable interpretability proxy for the overall model.

8.2 Text-GNN Fidelity

The Text-GNN Fidelity ($\boldsymbol{F}_{\mathbf{TG}}$) scores significantly increased across all models after applying TeGDA training, as seen in Table 3, compared to the original models in Table 2. Specifically, in the CommonsenseQA **IH**-test set, fidelity rose from 75.7% to 98.7% for QA-GNN, 40.7% to 98.0% for GreaseLM, and 51.0% to 95.0% for MHGRN. The original fidelity of QA-GNN was already higher (75.7%) than that of GreaseLM (40.7%) and MHGRN (51.0%), suggesting a more representative graph encoder in QA-GNN's architecture. Post-

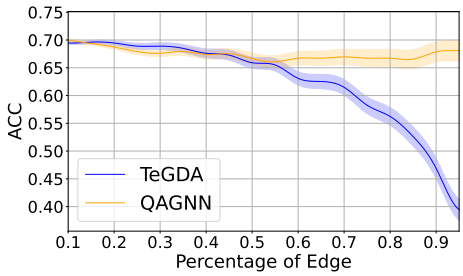

Figure 3: Accuracy on the **IH**-test set for original QA-GNN (orange) and trained with TeGDA (blue) as percentage of edges removed increases.

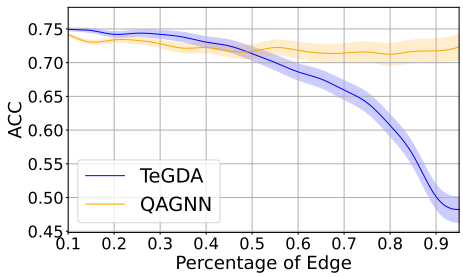

Figure 4: Accuracy on the **IH**-dev set for original QA-GNN (orange) and trained with TeGDA (blue) as percentage of edges removed increases.

TeGDA training, all models showed over 95% fidelity (F_{TG}), indicating a high consistency of graph encoder outputs with the original model outputs, despite masking the text encoder. This aligns with TeGDA's aim of enhancing explanation faithfulness. GreaseLM's fidelity notably improved from a low 40.7% to 98.0%, and achieved 99.6% on both OpenBookQA dev and test sets, demonstrating TeGDA's effectiveness in models with initially weak graph-text encoder connections.

8.3 Explanation Fidelity

For the lack of groundtruth GNN explanations, we will evaluate the obtained explanations in terms of fidelity (figure 3 and 4). To answer **Q3**, specifically, we sequentially remove edges from the GNN by following importance weight learned by the explanation model and test the classification performance. Generally, the removal of really important edges would significantly degrade the classification performance. Thus, a faster performance drop represents stronger fidelity. Figures 3 and 4 compare results on the test set and dev set respectively for the original QA-GNN model and QA-GNN trained with TeGDA. As more edges are removed, the accuracy of TeGDA drops much more rapidly compared to original QA-GNN. For example, after removing the top 1% of edges, the accuracy of the original QA-GNN remains relatively steady on both dev and test sets, while for TeGDA the accuracy drops 25-30%, indicating the explanations from TeGDA better capture the critical edges. The more rapid degradation for TeGDA as important edges are removed demonstrates its explanations have higher fidelity in reflecting the true reasoning process. This analysis provides quantitative evidence that the knowledge graph explanations extracted from the TeGDA model are more faithful.

8.4 Findings and Hypothesis

There are a few key differences that may explain why GreaseLM shows more improvement over the Text-GNN Fidelity metric under the TeGDA training scheme, and why the original QA-GNN may have achieved better graph fidelity compared to the others. (1) GreaseLM's separate graph encoder, focusing solely on the knowledge graph, may enhance the independence of LM and GNN structures, offering a balance between alignment and independence. This could explain its initial low fidelity, leading to high alignment accuracy later. In contrast, QA-GNN's integrated encoding of the QA context and knowledge graph could improve knowledge graph context incorporation but lacks a modulating control mechanism. Significantly, a discrepancy is observed between graph fidelity and text-graph consistency. MHGRN shows moderate fidelity in CommonsenseQA, suggesting that high fidelity doesn't guarantee high consistency. This underscores the importance of using both fidelity and consistency metrics for a comprehensive model evaluation.

9 Conclusion

Our study focused on assessing the faithfulness of knowledge graph explanations in commonsense reasoning models. We introduced Text-GNN Fidelity to evaluate the accuracy of these explanations. Our analysis revealed that the initial faithfulness of graph explanations in contemporary language model-knowledge graph (LM-KG) systems was limited. To address this, we developed a novel training method, Text-Graph Distribution-aware Alignment (TeGDA), which significantly enhances the consistency and fidelity of KG explanations, aligning them more closely with the actual reasoning processes of the models.

References

Guy Aglionby and Simone Teufel. 2022. Faithful knowledge graph explanations in commonsense question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10811–10817.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable ai for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459.

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)*, pages 1295–1309.

Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. 2019. Towards a deep and unified understanding of deep neural models in nlp. In *International conference on machine learning*, pages 2454–2463. PMLR.

Bernease Herman. 2017. The promise and peril of human evaluation for model interpretability. *arXiv preprint arXiv:1711.07414*.

Lynette Hirschman and Robert Gaizauskas. 2001. Natural language question answering: the view from here. *natural language engineering*, 7(4):275–300.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Shaobo Li, Xiaoguang Li, Lifeng Shang, Zhenhua Dong, Cheng-Jie Sun, Bingquan Liu, Zhenzhou Ji, Xin Jiang, and Qun Liu. 2022. How pre-trained language models capture factual knowledge? a causal-inspired analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1720–1732.

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)*, pages 2829–2839.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019a. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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*, pages 2381–2391.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832.

Xiaoman Pan, Kai Sun, Dian Yu, Jianshu Chen, Heng Ji, Claire Cardie, and Dong Yu. 2019. Improving question answering with external knowledge. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 27–37.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473.

H Ren, W Hu, and J Leskovec. 2020. Query2box: Reasoning over knowledge graphs in vector space using

box embeddings. In *International Conference on Learning Representations (ICLR)*.

Hongyu Ren, Hanjun Dai, Bo Dai, Xinyun Chen, Michihiro Yasunaga, Haitian Sun, Dale Schuurmans, Jure Leskovec, and Denny Zhou. 2021. Lego: Latent execution-guided reasoning for multi-hop question answering on knowledge graphs. In *International conference on machine learning*, pages 8959–8970. PMLR.

Hongyu Ren and Jure Leskovec. 2020. Beta embeddings for multi-hop logical reasoning in knowledge graphs. *Advances in Neural Information Processing Systems*, 33:19716–19726.

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.

Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. 2021. Interpreting graph neural networks for {nlp} with differentiable edge masking. In *International Conference on Learning Representations*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. 2022. Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5049–5060.

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, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, et al. 2019. Improving natural language inference using external knowledge in the science questions domain. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7208–7215.

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*, pages 535–546.

X Zhang, A Bosselut, M Yasunaga, H Ren, P Liang, C Manning, and J Leskovec. 2022. Greaselm: Graph reasoning enhanced language models for question answering. In *International Conference on Representation Learning (ICLR)*.

Tianxiang Zhao, Dongsheng Luo, Xiang Zhang, and Suhang Wang. 2023. Towards faithful and consistent explanations for graph neural networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 634–642.