XplainLLM: A Knowledge-Augmented Dataset for Reliable Grounded Explanations in LLMs

Zichen Chen[1] **Jianda Chen**[2] **Ambuj K. Singh**[1] **Misha Sra**[1]
[1]University of California, Santa Barbara
[2]Nanyang Technological University, Singapore
{zichen_chen, ambuj, sra}@ucsb.edu jianda001@ntu.edu.sg

Abstract

Large Language Models (LLMs) have achieved remarkable success in natural language tasks, yet understanding their reasoning processes remains a significant challenge. We address this by introducing XplainLLM, a dataset accompanying an explanation framework designed to enhance LLM transparency and reliability. Our dataset comprises 24,204 instances where each instance interprets the LLM's reasoning behavior using knowledge graphs (KGs) and graph attention networks (GAT), and includes explanations of LLMs such as the decoder-only Llama-3 and the encoder-only RoBERTa. XplainLLM also features a framework for generating grounded explanations and the *debugger-scores* for multidimensional quality analysis. Our explanations include *why-choose* and *why-not-choose* components, *reason-elements*, and *debugger-scores* that collectively illuminate the LLM's reasoning behavior. Our evaluations demonstrate XplainLLM's potential to reduce hallucinations and improve grounded explanation generation in LLMs. XplainLLM is a resource for researchers and practitioners to build trust and verify the reliability of LLM outputs. Our code and dataset are publicly available[1].

1 Introduction

As the capabilities and applications of large language models (LLMs) continue to expand (Liu et al., 2023; Achiam et al., 2023; Touvron et al., 2023; Jiang et al., 2024), the need for transparency and interpretability in their reasoning behavior has become increasingly urgent (Arrieta et al., 2020). Traditional methods (Ribeiro et al., 2016; Lundberg and Lee, 2017; Casalicchio et al., 2019) allow us to get insights into the reasoning behind language model outputs, but they fall short of providing a complete picture, leaving the logic behind complex decision opaque (Huang et al., 2023). This gap presents a significant barrier in applications where model decision transparency is important, such as healthcare (Ghosh et al., 2024), law (Cheong et al., 2024), and public services (Musumeci et al., 2024).

Current methods for explaining LLM's reasoning behavior primarily focus on the analysis of parameter changes (Clark et al., 2019; Jacovi et al., 2021; Bills et al., 2023) and chain-of-thought (CoT) based self-explanation (Huang et al.; Li et al., 2023). Analysis of parameter changes bases the explanations on self-attention weights in models like BERT (Kenton and Toutanova, 2019) and GPT-2 (Radford et al., 2019), deducing correlations between input tokens and the model's predictions. However, the relationships highlighted in these generated explanations are difficult to understand for humans. CoT-based self-explanation, on the other hand, iteratively generates rationales step-by-step. Due to the inherent constraints in LLMs, these explanations often have hallucinations and can not reflect the real reasoning process (Huang et al., 2023).

We introduce XplainLLM, a dataset accompanying an explanation framework designed to enhance transparency, explainability, and understandability in LLM reasoning behaviors. By integrating knowledge graphs (KGs) and Graph Attention Networks (GAT) (Veličković et al., 2018), we construct a structured and reliable dataset that anchors explanations in reasoning-relevant knowledge. We link the LLM reasoning process to the entities and relations within KGs to help provide an intuitive and interpretable representation of the LLM's decision-making process. Our process also helps facilitate model tuning, debugging, robustness evaluation and demonstration in in-context learning. XplainLLM provides a structured explanation of two distinct types of LLMs: Llama-3-8B (Touvron et al., 2023) (decoder-only model) and RoBERTa-large (Liu et al., 2019) (encoder-only model). A total of 24,204 instances are included in the dataset. The explanations are tied to two models' reasoning

[1]https://lmexplainer.github.io/xplainllm

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 7578–7596
November 12-16, 2024

Dataset	Size	Answer Format	Expl. Format	Source	Model Match?	Self-Explanatory?	"Why Not" Included?
CoS-E	9,500	MC	NL	Human	×	×	×
ECQA	10,962	MC	NL	Human	×	✓	×
Neuron	307,200	Neuron	NL + Score	Model	✓	×	×
`XplainLLM`	24,204	MC	NL	Model	✓	✓	✓

Table 1: Comparison of prevalent explanation datasets with `XplainLLM`, detailing instance count (Size), answer types (Answer Format: e.g., multiple-choice (MC)), explanation styles (Explanation Format: e.g., natural language (NL)), origin (Source), alignment with model reasoning (Model Match?), necessity of human intervention to deduce the reasoning (Self-Explanatory?), and inclusion of reasons for alternative answer rejection "Why Not" Included?).

processes, derived from their performance on the CommonsenseQA (Talmor et al., 2019) challenge.

Additionally, we introduce an explanation framework that utilizes a retrieval-based method to support generating grounded explanations for LLMs. This framework operates without the need for additional model training, utilizing `XplainLLM` as a knowledge base to retrieve the most relevant data points to the given query. The selected data points serve as demonstration examples for in-context learning (Dong et al., 2022), enabling the LLMs to generate explanations that are more grounded in the reasoning process.

We evaluate the quality of the explanations in `XplainLLM` through human and automated evaluations. The overall quality of explanations achieves an average score of 0.87/1.00 by human evaluators, and an average of 0.89/1.00 by automated evaluators. We evaluate our framework by comparing the performance of LLMs with and without our framework, and the results show that LLMs under our framework outperform the benchmark, with a performance gap extending to 20%. We also evaluate the quality of the explanations generated by our framework, and the results underscore the quality of our explanations on multiple metrics.

In summary, we make two key contributions to the field of explainable AI for LLMs: (1) an explanation dataset of model reasoning behavior, and (2) a framework for improving the interpretability of LLMs through structured, grounded explanations. To the best of our knowledge, `XplainLLM` is the first dataset to provide structured and grounded explanations for LLM reasoning behavior.

2 Related Work

Interpretability in LLMs Explainable AI (XAI) aims to address the issue of interpreting the outcomes of language models (Li et al., 2023; Wiegreffe et al., 2021; Madsen et al., 2022). One of its goals is to generate explanations that enable humans to easily understand the decision-making process. Zelikman et al. (2022); Zhang and Gao (2023); Wang et al. (2023) utilize gradual strategies that iteratively generates the rationales step-by-step. Huang et al.; Chen et al. (2023a); Tanneru et al. (2024); Chakraborty et al. (2023) utilize the CoT to find the rationale and apply the reasoning capabilities of LLMs to domain tasks. However, these explanations are inherently constrained in capturing prompt-specific reasoning, which often generates hallucinations and can not reflect the real reasoning of LLMs (Turpin et al., 2024).

Another goal of XAI is to explain the model in a trustworthy way. Rajani et al. (2019a) introduce an explainable factor to minimize the risk of unreasonable explanation generation. Chen et al. (2021) integrate the external knowledge to generate why and why-not counterfactual explanations. Zelikman et al. (2022) apply self-checker mechanism to ensure trusted rationals. However, these methods fail to accurately capture the core reasoning of LLMs. In contrast, our work enhances LLM trustworthiness and deepens human understanding of its reasoning behavior, improving their potential in end-user applications.

Explanation Datasets The explainable datasets for language models can be categorized into three types (Wiegreffe and Marasovic, 2021): (1) highlights: provide input elements such as words and phrases, as explanations to a predicted output (Camburu et al., 2018; DeYoung et al., 2020; Yin et al., 2021; Bills et al., 2023); (2) free-text explanations: provide readable textual explanations in words or sentences (Rajani et al., 2019b; Sap et al., 2020; Brahman et al., 2021); (3) structured explanations: provide natural language explanation but are constrained by the explanation writing process (Aggarwal et al., 2021; Jhamtani and Clark, 2020; Inoue et al., 2020). Different from these, our explanation incorporates highlighted reason-elements and guided instruction to generate a free-text explana-

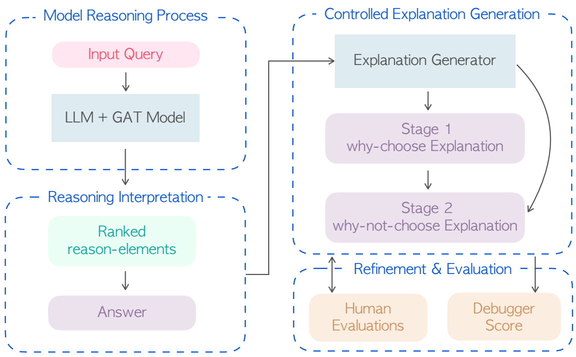

Figure 1: Overview of XplainLLM in LLM Reasoning Interpretation and Explanation Generation. XplainLLM integrates LLM with GAT to interpret the reasoning process and generate explanations. It consists of two stages: "*why-choose*" and "*why-not-choose*" explanations. The explanations are evaluated by human feedback and debugger scores.

tion. Our explanation is structured and grounded in the reasoning process, enhancing the trustworthiness and comprehensiveness of the content. We present a comparison with prevalent explanation datasets (Rajani et al., 2019b; Aggarwal et al., 2021; Bills et al., 2023) in Table 1.

3 XplainLLM: Dataset, Explanation Framework and Debugger-Score

XplainLLM serves three essential purposes in interpreting LLMs' reasoning behavior. First, it utilizes KG and GAT to interpret LLM through parameter changes (Chen et al., 2023b), collecting these explanations to build a dataset. The LLMs we used are Llama-3-8B (Touvron et al., 2023) (decoder-only) and RoBERTa-large (Liu et al., 2019) (encoder-only). Second, we provide an explanation framework for generating faithfully grounded explanations without additional training. Third, we introduce the *debugger-score*, which is designed for multidimensional analysis to quantify the quality of explanations, supporting our framework for comprehensively evaluating and improving LLM explainability.

3.1 Task Definition and Collection Method

The primary goal of XplainLLM is to enhance the interpretability of LLMs through grounded explanations. We define the task as generating explanations that clarify the decision-making processes behind model predictions. We use QA tasks to generate instances for our dataset. The overview of the collection process is shown in Figure 1. A more detailed data collection description is shown in Appendix G.

The LLM's reasoning is grounded in a structured KG, which is used to identify the most salient features that influence the model's predictions. We employ GAT to analyze the KG's structure and identify the influence of specific nodes and edges that are salient to the model's decision-making process. Each instance in XplainLLM is formulated as follows:

$$\text{Instance} = ((Q, A), \text{Explanation}) \tag{1}$$

where (Q, A) is the question-answer pair and Explanation includes:

- A *why-choose* explanation, detailing the reason behind the model's answer choice.
- A *why-not-choose* explanation, detailing reasons against alternative choices.
- *Ranked reason-elements*, identified through GATs that analyze the KG's structure to identify critical influencing elements.
- A *debugger-score* for each explanation, quantifying its faithfulness, completeness, accuracy and overall quality.

Graph-Based Reasoning Interpretation. To produce the aforementioned explanation, we introduce a graph-based interpreting method to learn the features that influence the model's decision-making process. We first extract the key elements from the KG g. The criteria for selecting KG are based on its coverage, quality, and relevance to the task. KG selection is based on comprehensiveness, quality, and task relevance. The extraction process involves identifying nodes and edges within the g that are relevant to the input question and answer pair. We incorporate node relevance scores into this retrieval process, using the LLM's knowledge to guide the pruning of the g:

$$G_e = \text{PruneKG}(Q, A, g, s_i) \tag{2}$$

where s_i represents the relevance score for each node i in the retrieved graph, calculated using LLM's probability function that assesses the alignment of node embeddings with the input context (Q, A). The function PruneKG evaluates the semantic relationship between node embeddings and the query. This extraction leverages the LLM's knowledge to focus on the most informative elements for the given QA context. The algorithm for constructing the G_e is provided in Appendix A.

Once the relevant subgraph G_e is obtained, we use a GAT to determine the significance of each node and edge in contributing to the model's output. Each node i at k-th layer is represented by a feature vector h_{ki}. The attention α_{ij} for each node pair (i, j) are computed using a softmax function over a parameterized self-attention mechanism a that captures the relationship dynamics:

$$\alpha_{ij} = \frac{\exp(a(h_{ki}, h_{kj}))}{\sum_{l \in \mathcal{N}(i)} \exp(a(h_{ki}, h_{kl}))} \quad (3)$$

where $\mathcal{N}(i)$ denotes the neighbors of node i.

The updated node features $h_{k+1,i}$ are computed by aggregating the features of neighboring nodes weighted by their respective attention scores:

$$h_{k+1,i} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W f_m(h_{kj}, u_i, r_{ij})\right) + h_{ki} \quad (4)$$

where f_m is a multi-layer perceptron (MLP) that processes features of neighboring nodes considering their types and interrelations. W is a weight matrix, σ is a non-linear activation function. We provide the details of the GAT model in Appendix B.

We define the probability of selecting an answer v from the set A by leveraging both the representation embeddings from the language model ($\mathbf{H}^{LM}$) and the graph-based reasoning features (h_K and α_K) extracted from our subgraph G_e:

$$P(a|q) \propto \exp(\text{MLP}(\mathbf{H}^{LM}, h_K, \alpha_K)) \quad (5)$$

where h_K represents the output features from the final layer of our K-layer graph reasoning network, and α_K represents the attention coefficients. To this end, we map the LLM's reasoning to the graph features. The extracted attention features are mapped to their corresponding nodes in the G_e, and we select the top n nodes with the highest attention scores for generating the explanations.

Controlled Explanation Generation. Upon obtaining the reasoning features, we transform them into structured and human-understandable explanations through a two-stage instructional process. The top n nodes are selected as the key *reason-elements* set R, which guides the explanation generator model $\mathbb{F}$ to construct the explanations. The explanation generation process includes: (1) *why-choose* explanation: the reasoning behavior behind the model's choice, and (2) *why-not-choose* explanation: the rationale for dismissing other potential answers. The instruction for the *why-choose* stage is:

$$\begin{aligned} &Basis : [TASK\_TYPE], \\ &Input : [Q, A], \\ &Output : [y', R], \\ &Explanation(S\_1) : [y']. \end{aligned}$$

The output of stage 1 named E_{why} is used as the input for stage 2. The instruction for *why-not-choose* stage is:

$$Explanation(S\_2) : [E_{why},\ A \setminus \{y'\}].$$

The details of the instruction are provided in the Appendix C.

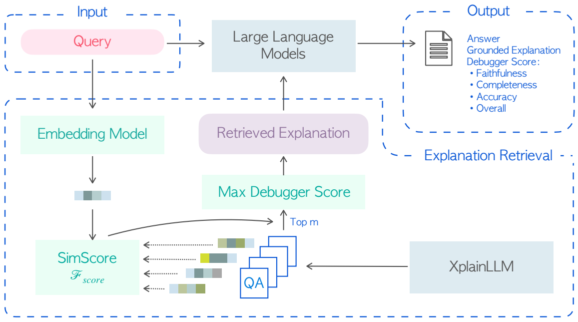

Figure 2: Explanation Framework for Grounded Explanation Generation in LLMs. An embedding model is used to compute query embeddings, and `XplainLLM` retrieves the top m instances based on $SimScore \mathcal{F}_{\text{score}}$. The explanation with the highest *debugger-score* from each instance is selected to generate grounded explanations.

3.2 Explanation Framework for Grounded Explanations

To enhance the usability of `XplainLLM` and facilitate the generation of grounded explanations for different types of LLMs (especially for private LLMs, e.g., GPT-4), we introduce an explanation framework that leverages the collected dataset to generate faithfully grounded explanations without additional model training. The framework is illustrated in Figure 2. The process is divided into three steps:

Embedding Calculation. When receiving a new query $(Q_{\text{new}}, A_{\text{new}})$, its embeddings $\mathbf{e}_{QA_{\text{new}}}$ is calculated using the an embedding models. To generalize our framework, we use voyage-2-large model from VOYAGE AI[2], as our embedding model to extract the embeddings, due to its state-of-the-art performance in generalist text embedding[3].

[2] https://docs.voyageai.com/docs/embeddings
[3] https://huggingface.co/spaces/mteb/leaderboard

Similarity Computation and Retrieval. We retrieve the most contextually relevant instances by computing the cosine similarity between new query embedding $\mathbf{e}_{QA_{new}}$ and instance embedding $\mathbf{e}_{QA}$ in `XplainLLM` $\mathcal{E}$:

$$\mathcal{F}_{\text{score}}(\mathbf{e}_{QA_{new}}, \mathbf{e}_{QA}) = \frac{\mathbf{e}_{QA_{new}}^{\top}\mathbf{e}_{QA}}{\|\mathbf{e}_{QA_{new}}\|_2\|\mathbf{e}_{QA}\|_2}$$

This function $\mathcal{F}_{\text{score}}$ scores each instance $\text{sim}(\mathbf{e}_{QA_{new}}, \mathbf{e}_{QA})$ for relevance. The $\mathbf{e}_{QA}$ can be pre-computed and stored in the dataset for efficient retrieval. To accelerate the retrieval process, we provide embeddings for each instance in `XplainLLM`, using voyage-2-large.

Instance Selection and Explanation Generation. The top m instances with the highest similarity scores, $\text{sim}(\mathbf{e}_{QA_{new}}, \mathbf{e}_{QA})$, are selected. Each instance may contain multiple explanations from different LLMs, denoted as $\mathcal{E}_t$, where t indexes the instances. For each instance, we select the explanation e^* that maximizes the *debugger-score* set D:

$$e^* = \arg\max_{e \in \mathcal{E}_t} \sum_{d \in D} w_d \cdot D(e, d)$$

where and w_d are the weights reflecting user preferences for each dimension. This selection is influenced by user-specified preferences which dictate the importance of various dimensions of explanation quality, such as faithfulness or accuracy. We will introduce the *debugger-score* in Section 3.3. These selected instances are used as in-context learning examples for targeted LLM to generate grounded explanations.

3.3 Debugger-Score for Explanation Analysis

To improve the understanding of generated explanations, we introduce the *debugger-score* to evaluate the quality of explanations. Inspired by the method of transformer debugging (Bills et al., 2023), our *debugger-score* simulates a "perfect" LLM to benchmark against the actual LLM's reasoning. It quantifies the quality of explanations by assessing:

1. **Faithfulness:** How accurately the explanations reflect the actual reasoning of the LLM.
2. **Completeness:** Whether the explanations cover all essential aspects of the reasoning process.
3. **Accuracy:** The correctness of the explanation in terms of factual and contextual relevance.
4. **Overall:** The overall quality of the explanation, combining the above dimensions.

	Why-choose	Why-not-choose	Whole Explanation
Overall	94.77	85.74	180.81
Training Set	94.41	85.22	179.63
Dev Set	93.00	84.54	178.44
Testing Set	96.89	87.46	184.35

Table 2: The average word counts of *why-choose* explanation, *why-not-choose* explanation and whole explanation in our `XplainLLM` dataset.

The *debugger-score* utilizes predefined instructions to guide the evaluation, focusing on identifying discrepancies between the simulated "perfect" LLM and the actual LLM. Our evaluation method quantifies the quality of explanations, providing a measure of where the LLM's reasoning succeeds or falls short. Our *debugger-score* is used to enhance the reliability and transparency of the explanations. Further details on the implementation, functionality, and faithfulness evaluation of the *debugger-score* can be found in the Appendix E.

4 Dataset Overview and Preparation

4.1 Dataset Description

Schema. `XplainLLM` contains fields that correspond to the QA pair, the model's predicted answer, the ground-truth label, and an explanation set.

Explanations Set. The explanation set includes a set of 50 *reason-elements*, e.g., words or phrases, sorted by attentions, a set of top-ranked *reason-elements*, a *why-choose* explanation in free-text form, a *why-not-choose* explanation also in free-text form. Example instances are shown in Appendix D.

Statistics. `XplainLLM` includes 24,204 instances of explanations, split according to the official CommonsenseQA's partitioning into three sets: the training, development (dev), and testing sets. The average word count of E_{why} and $E_{why-not}$ are 94.77 and 85.74 respectively, resulting in an aggregate count of approximately 180.81 words per whole explanation. A more detailed breakdown of the average word count is provided in Table 2. Additional statistics can be found in Appendix I.

4.2 Data Preparation

`XplainLLM` captures and analyzes the reasoning behavior of LLMs on CommonsenseQA dataset (Talmor et al., 2019). CommonsenseQA serves as a foundational benchmark for assessing the commonsense reasoning capabilities of these models.

We select Llama-3-8B and RoBERTa-large as LLMs for our dataset as they exemplify decoder-only and encoder-only LLMs respectively, providing a comprehensive view of different model architectures in language understanding. The models are fine-tuned on CommonsenseQA's official training set, to understand and interpret the complexities of commonsense reasoning. We utilize ConceptNet (Speer et al., 2017) as our KG to obtain g_e, due to its extensive coverage of commonsense knowledge. It contains over 13 million links between concepts and their interrelations, providing a rich source of general knowledge. We use a 5-layer GAT model to extract the reasoning paths. We use GPT-3.5-turbo (Ouyang et al., 2022) and GPT-4-turbo (Achiam et al., 2023) as explanation generator model $\mathbb{F}$ to generate a natural language explanation in a sentence or a paragraph. To ensure the quality of our dataset, we conduct a post-generation evaluation. All explanations undergo human review. Human evaluators identify inaccuracies, and any discrepancies in explanations, and return to $\mathbb{F}$ for refinement. This procedure mitigates potential issues from model-generated explanations, guaranteeing clarity and relevance aligned with human understanding. We also provide embeddings of the (Q, A) pair for each instance in the dataset. The embeddings are generated using the voyage-large-2. The *debugger-score* is calculated using GPT-4-turbo. Further experiment specifics and data collection procedures are provided in the Appendix F and G.

5 Experiments and Evaluation

5.1 Evaluation Methodology

We evaluate `XplainLLM` and explanation framework through two main perspectives:

1. **Explanation Quality Evaluation:** The quality of the explanations generated by the LLMs is assessed via a dual approach: **(1) Human Evaluation** - Experts and crowdsourcing review the explanations, and **(2) Automated Evaluation** - GPTs evaluate the explanations.

2. **Framework Effectiveness:** We measure the impact of our proposed methods on the groundedness of newly generated explanations and the performance of the LLMs. This includes: **(1) Grounded Explanation Assessment** - Using the *debugger-score* to evaluate how well the explanations are grounded in factual content, and **(2) Performance Analysis** - We evaluate changes in the accuracy of the LLM outputs by comparing metrics before and after applying our framework.

	Expert <-> GPT-3.5	Expert <-> GPT-4	GPT-3.5 <-> GPT-4
ρ	0.70	0.60	0.66

Table 3: Correlation coefficient (ρ) between overall quality scores evaluated by expert, GPT-3.5 and GPT-4.

Specifically, the evaluation metrics for explanation quality assessment are human-centered metrics, following the guidelines of Hoffman et al. (2018). Each explanation is assessed using seven evaluative questions that explore different aspects of the explanation's impact and quality. The metrics encompass overall quality, understandability, trustworthiness, satisfaction, detail sufficiency, completeness, and accuracy. Evaluators allocate scores to these questions using a three-point Likert scale: 1 (disagree), 2 (neutral), and 3 (agree). Subsequently, scores are normalized to the range [0, 1]. Higher scores suggest better quality. Detailed definitions are provided in the Appendix J.2.

5.2 Explanation Quality Evaluation

We conducted human and automated evaluations to go beyond the technical evaluation of the explanations. The human evaluation involved three experts with NLP backgrounds and 50 general users via Prolific[4]. Our participant pool was gender-balanced, and comprised of native English speakers with at least a high school education. Experts and users rate 20 randomly selected explanations based on guidelines adapted from (Hoffman et al., 2018) to ensure consistency and mitigate bias. Automated evaluations are performed using GPT-3.5-turbo and GPT-4 to parallel human judgment, quantifying performance with standardized scores. Detailed methodologies and participant instructions are provided in Appendix J.1.

Results of Expert and Automated Evaluation. The feedback from human experts highlighted the distinctiveness of our explanations compared to existing methods. One expert remarked,

> "*In comparison to prior explanations, these explanations provide a more intu-*

[4] `https://www.prolific.com`

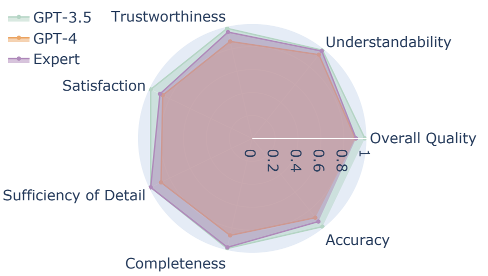

Figure 3: Evaluation by human experts, automated evaluator GPT-3.5 and GPT-4.

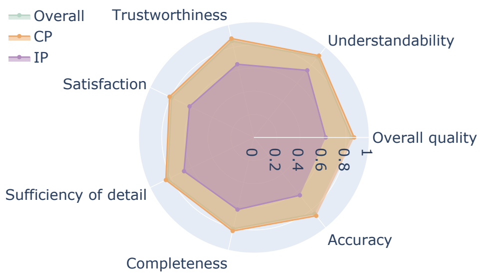

Figure 4: Human evaluation of explanations: Overall, CP, and IP. Note that the CP scores align closely with the overall scores.

> *itive understanding of the LLM's reasoning behavior. The explanations are cogent, and even in instances of erroneous predictions, the underlying reasoning remains transparent and comprehensible.*"

This feedback underscores the clarity and transparency of our explanations.

The results are summarized in Table 4. Human experts assign an average score of 0.93 across seven evaluation metrics, with "understandability" and "completeness" receiving the highest scores. The automated evaluators, GPT-3.5 and GPT-4, assign average scores of 0.91 and 0.92, respectively. The performance of these automated evaluators aligns closely with human expert evaluations across dimensions, as shown in Figure 3.

Further insights into the human-like understanding of automated evaluators and their assessment of explanations are detailed in Table 3. This data shows a significant agreement between the automated evaluators and human experts. Such findings further support the credibility and value of our explanations.

Results of Crowdsourcing Evaluation. we present the average scores from crowdsourcing on eight metrics, as depicted in Figure 4. These scores reflect evaluations of the overall explanations, as well as separate assessments for explanations of correct predictions (CP) and incorrect predictions (IP). The details of our analysis are discussed below.

Participants assigned a high average score of 0.87 to the overall quality of our explanations, indicating a favourable perception and underscoring their above-average clarity. The explanations received an average understandability score of 0.89, demonstrating their clarity. The low variance of 0.26 suggests consistent comprehension among participants. However, a detailed analysis shows a disparity based on the LLM's prediction accuracy: explanations for correct predictions (CP) were highly rated at 0.91 with a variance of 0.26, while explanations for incorrect predictions (IP) scored lower at 0.74 with a variance of 0.65, indicating less clarity and greater variability in participant responses.

In terms of trustworthiness, our explanations scored an average of 0.88 for CP. A Pearson correlation coefficient of 0.71 between trustworthiness and understandability confirms a strong positive relationship, suggesting that clearer explanations enhance participants' trust in the LLM's outputs.

Overall satisfaction with our explanations is high, with 86% of participants stating that the explanations meet or exceed their expectations. 97.36% of the explanations are considered sufficiently detailed. The completeness of our explanations also received high marks, with an average score of 0.81 and a median score of 1.00, suggesting that over half of the participants find the explanations to be entirely comprehensive. However, the distribution may reflect differences in the evaluators' familiarity with AI or occasional oversimplifications by the model. The accuracy of the explanations are rated at 0.84, with a noticeable disparity between CP at 0.87 and IP at 0.64, highlighting how the LLM's prediction accuracy significantly influences the perceived accuracy of explanations. Furthermore, a Pearson correlation of 0.68 between accuracy and trustworthiness indicates that more accurate explanations are considered more trustworthy.

The positive feedback from our crowdsourcing evaluations robustly validates `XplainLLM`, demonstrating its effectiveness in conveying the complexities of the LLM's decision-making in a clear, trustworthy, and satisfying manner to users.

	Overall quality	Understandability	Trustworthiness	Satisfaction	Sufficiency of detail	Completeness	Accuracy
GPT-3.5	0.98	0.98	0.98	0.98	0.98	0.98	0.98
GPT-4	0.90	0.93	0.87	0.87	0.88	0.87	0.88
Human Expert	0.91	0.97	0.95	0.89	0.98	0.97	0.93
Crowdsourcing	0.85	0.89	0.86	0.80	0.83	0.81	0.85

Table 4: Evaluation by automated evaluator GPT-3.5, GPT-4, human experts and crowdsourcing, on seven evaluation metrics.

Model	#P	Version	Faithfulness	Completeness	Accuracy	Overall
gpt-3.5-turbo	Unknown	Vanilla	**0.70**	0.59	**0.73**	**0.67**
		`XplainLLM`	0.69	**0.61**	0.71	0.67
gpt-4-turbo	Unknown	Vanilla	0.73	0.62	0.79	0.71
		`XplainLLM`	**0.81**	**0.73**	**0.82**	**0.79**
llama3-8b	8.02B	Vanilla	**0.70**	0.53	**0.72**	**0.65**
		`XplainLLM`	0.67	**0.57**	0.67	0.64
llama3-70b	70.6B	Vanilla	0.72	0.57	0.76	0.68
		`XplainLLM`	**0.79**	**0.62**	**0.81**	**0.74**
mixtral-8x7b	46.7B	Vanilla	0.74	0.58	0.76	0.69
		`XplainLLM`	**0.75**	**0.61**	**0.76**	**0.71**

Table 5: Comparison of Vanilla and `XplainLLM` Versions of Models with *debugger-score*.

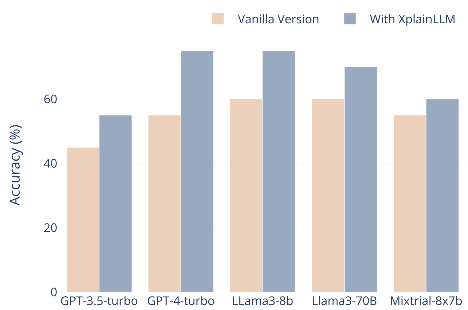

Figure 5: Accuracy comparison of vanilla version and with `XplainLLM` version for different models. `XplainLLM` consistently improves accuracy across all models.

5.3 Framework Evaluation

In evaluating our proposed framework, we include five LLMs: GPT-3.5-turbo (Brown et al., 2020), GPT-4-turbo (Achiam et al., 2023), Llama3-8B (Touvron et al., 2023), Llama3-70B (Touvron et al., 2023), and Mixtral-8x7B (Jiang et al., 2024). These models are selected due to their diversity in scale, architecture, and popularity in open-source research. Our goal is to evaluate the generalizability and scalability of our framework across different LLMs without additional training. We compare the vanilla versions of these models with the versions enhanced by `XplainLLM`. The results are summarized in Table 5. We specifically selected 20 questions from `XplainLLM` designed to challenge models by exposing their tendency to produce hallucinations. These questions are carefully chosen to test the framework's ability to generate grounded explanations. We then evaluate five LLMs both with and without the enhancements provided by `XplainLLM`, allowing us to explore how our framework performs across different scales and architectures. The benchmarks for this evaluation are focused on four key metrics: *faithfulness*, *completeness*, *accuracy*, and *overall performance*, as shown in Table 5. All values are normalized to a scale of 1.0. We further quantified the impact of our framework by comparing the accuracy rates of the vanilla version to those enhanced with our modifications, as detailed in Figure 5.

Our results show that performance variations across different model architectures and configurations, as demonstrated in Table 5. Notably, the GPT-4-turbo model, when enhanced with our framework, demonstrates exceptional performance across key metrics. It scores 0.81 in Faithfulness, 0.73 in Completeness, and 0.82 in Accuracy, culminating in an Overall score of 0.79. These high scores suggest that our framework not only improves the overall output quality but also ensures that the LLM's reasoning is grounded in faithful knowledge, thus enhancing both the clarity and reliability of the model's behavior explanation. We notice that models like GPT-3.5-turbo and Llama-3-8B exhibit suboptimal results in certain cases, likely due to limitations in their inherent ability to generalize with in-context examples (Xu et al., 2024). Further studies with different in-context example sizes are detailed in Appendix F.2. The study results show that some models can be sensitive to the number of in-context examples, which affects the quality of their generated explanations. We suggest optimal in-context configurations to be model-specific to fully leverage our framework's benefits.

We also observe a consistent improvement in accuracy across different LLMs when our framework is applied, as shown in Figure 5, which implies a scalable utility of our framework. We find the GPT-4-turbo model exhibits the most significant improvement. This may suggest that our enhancements are effective in assisting more complex LLMs to ground their reasoning in faithful knowledge, thereby reducing hallucinations and improving interpretability.

By comparing the detailed reasoning explanation of the models with and without our framework, we observe that the explanations generated under the vanilla models tend to generate outputs that are not entirely supported by input data, leading to hallucinations. We provide a comparison example in Appendix H.

These results demonstrate our framework can guide the LLMs toward a more grounded and data-driven approach in generating outputs. This is helpful for applications where precision and reliability are paramount, such as in legal, medical, or safety-critical environments. Furthermore, the consistent improvements across LLMs of varying capabilities suggest that our framework is robust and scalable, capable of enhancing a wide range of AI systems. This broad applicability suggests potential for widespread adoption in enhancing the transparency and accountability of AI decision-making processes.

6 Conclusion

We introduce `XplainLLM`: a knowledge-augmented dataset paired with an explanation framework designed to enhance the interpretability of LLMs. Our dataset and framework provide a way for LLMs to generate reliable and grounded explanations without additional training. Through the use of *debugger-score*, we provide a multidimensional analysis of quantitatively evaluate the quality of explanations. Our evaluations demonstrate that `XplainLLM` not only grounds explanations in reasoning behavior, but also helps LLMs reduce hallucinations and improve their performance. The dataset and code are available at https://lmexplainer.github.io/xplainllm. We release them under the MIT license to encourage further research in explainable AI.

Limitation

Committed to transparency and rigorous analysis, we acknowledge potential limitations in our dataset. Since our *reason-elements* R is originally derived from g_e, any inherent limitations or inaccuracies within used KG could influence the quality of our explanations.

Ethical Considerations

While `XplainLLM` and its accompanying explanation framework provides advancements in the transparency and accountability of LLMs, several risks might exist. First, the reliance on KGs and structured data may lead to biases embedded in these sources, potentially skewing the explanations. Secondly, incorrect knowledge augmentation could mislead users about the accuracy of the explanations. Additionally, there is a risk that users might over-rely on the *debugger-score* without critical assessment, potentially overlooking context-specific inaccuracies. It is essential for future work to continuously refine `XplainLLM`, address detected biases, and enhance the robustness of the framework to mitigate these risks.

Acknowledgement

We thank Mitali Gaidhani and Kirill Aristarkhov for their help with questionnaire data collection, Junrui Liu for paper revisions, and Bairu Hou for insightful discussions. This project was partially supported by funding from the National Science Foundation under grant #IIS-2229876.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for commonsenseqa: New dataset and models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Faeze Brahman, Vered Shwartz, Rachel Rudinger, and Yejin Choi. 2021. Learning to rationalize for non-monotonic reasoning with distant supervision. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12592–12601.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Giuseppe Casalicchio, Christoph Molnar, and Bernd Bischl. 2019. Visualizing the feature importance for black box models. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pages 655–670. Springer.

Saikat Chakraborty, Shuvendu Lahiri, Sarah Fakhoury, Akash Lal, Madanlal Musuvathi, Aseem Rastogi, Aditya Senthilnathan, Rahul Sharma, and Nikhil Swamy. 2023. Ranking llm-generated loop invariants for program verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9164–9175.

Qianglong Chen, Feng Ji, Xiangji Zeng, Feng-Lin Li, Ji Zhang, Haiqing Chen, and Yin Zhang. 2021. KACE: Generating knowledge aware contrastive explanations for natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2516–2527, Online. Association for Computational Linguistics.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023a. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*.

Zichen Chen, Jianda Chen, Yuanyuan Chen, Han Yu, Ambuj Singh, and Misha Sra. 2023b. Lmexplainer: a knowledge-enhanced explainer for language models. *arXiv preprint arXiv:2303.16537*.

Inyoung Cheong, King Xia, KJ Kevin Feng, Quan Ze Chen, and Amy X Zhang. 2024. (a) i am not a lawyer, but...: Engaging legal experts towards responsible llm policies for legal advice. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 2454–2469.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Akash Ghosh, Arkadeep Acharya, Raghav Jain, Sriparna Saha, Aman Chadha, and Setu Sinha. 2024. Clipsyntel: clip and llm synergy for multimodal question summarization in healthcare. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22031–22039.

Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. 2018. Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*.

Shiyuan Huang, Siddarth Mamidanna, Shreedhar Jangam, Yilun Zhou, and Leilani H Gilpin. 2023. Can large language models explain themselves? a study of llm-generated self-explanations. *arXiv preprint arXiv:2310.11207*.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *6th Annual Conference on Robot Learning*.

Naoya Inoue, Pontus Stenetorp, and Kentaro Inui. 2020. R4C: A benchmark for evaluating RC systems to get the right answer for the right reason. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6740–6750, Online. Association for Computational Linguistics.

Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. 2021. Contrastive explanations for model interpretability. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1597–1611.

Harsh Jhamtani and Peter Clark. 2020. Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 137–150, Online. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. 2023. Trustworthy ai: From principles to practices. *ACM Computing Surveys*, 55(9):1–46.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

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*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, 55(8):1–42.

Emanuele Musumeci, Michele Brienza, Vincenzo Suriani, Daniele Nardi, and Domenico Daniele Bloisi. 2024. Llm based multi-agent generation of semi-structured documents from semantic templates in the public administration domain. In *International Conference on Human-Computer Interaction*, pages 98–117. Springer.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019a. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019b. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online. Association for Computational Linguistics.

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.

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.

Sree Harsha Tanneru, Chirag Agarwal, and Himabindu Lakkaraju. 2024. Quantifying uncertainty in natural language explanations of large language models. In *International Conference on Artificial Intelligence and Statistics*, pages 1072–1080. PMLR.

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

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2024. Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36.

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

Boshi Wang, Xiang Yue, and Huan Sun. 2023. Can chatgpt defend its belief in truth? evaluating llm reasoning via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11865–11881.

Sarah Wiegreffe and Ana Marasovic. 2021. Teach me to explain: A review of datasets for explainable natural language processing. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Sarah Wiegreffe, Ana Marasović, and Noah A Smith. 2021. Measuring association between labels and free-text rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284.

Xin Xu, Yue Liu, Panupong Pasupat, Mehran Kazemi, et al. 2024. In-context learning with retrieved demonstrations for language models: A survey. *arXiv preprint arXiv:2401.11624*.

Kayo Yin, Patrick Fernandes, Danish Pruthi, Aditi Chaudhary, André F. T. Martins, and Graham Neubig. 2021. Do context-aware translation models pay the right attention? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 788–801, Online. Association for Computational Linguistics.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STar: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*.

Xuan Zhang and Wei Gao. 2023. Towards llm-based fact verification on news claims with a hierarchical step-by-step prompting method. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1011.

Appendix

A Graph Construction Algorithm

Algorithm 1: Sub-graph Construction (PruneKG)

Data: Graph g with nodes n, input content QA, encoding function of LLM f_{enc}, MLP f_s^{node}, Number of top nodes to select N

Result: Pruned graph g_e

begin
  Initialize an empty list *node_scores* ;
  for *each node n in g* **do**
    Obtain the embedding of n: $\mathcal{B} \leftarrow f_{enc}(n||QA)$;
    Compute the relevance score of n: $s_i \leftarrow sigmoid(f_s^{node}(\mathcal{B}))$;
    Append (n, s_i) to *node_scores* ;
  end
  Sort *node_scores* in descending order based on s_i ;
  Select the top N nodes from the *node_scores* list ;
  Create a new graph g_e with the selected L nodes, preserving their edges and properties ;
  return g_e ;
end

B Details of Graph Attention Network

In Section 3.1, we detail the method for interpreting the LLM's reasoning behavior through graph-based techniques. We provide supplementary calculations and algorithmic details in this section.

We describe the process for updating the node features in a graph using a GAT in Equation (4). Here, each node i updates its feature vector $h_{k+1,i}$ based on the features of its neighboring nodes $N(i)$. f_m is transformation function, modeled as a MLP, that maps the input features h_{kj}, u_i , and r_{ij} into a higher-dimensional space. specifically, u_i is the one-hot vector encoding the type of node i, and r_{ij} is the relation embedding denoting the relationship type between nodes i and j, calculated by:

$$r_{ij} = f_\theta(i, u_{ij}) = f_\theta(i, u_i \parallel u_j), \tag{6}$$

where u_{ij} is an one-hot vector encoding the type of connection between nodes i and j, and u_{ij} is the concatenation of i and j.

C Instruction for Explanation Generation

Due to the space constraints, we provide detailed guidelines and instructions for generating explanations in this section.

Basis: Given a LM augmented with a graph attention network to extract key reasoning elements for decision-making. The task is [TASK_TYPE].

Input: The question is: $[Q]$. The Answer Options are: $[A]$

Output: The model predicted choice $[y']$. Based on the Ranked Reason-elements: $[\mathcal{R}]$

Explanation (Stage 1): Explain the LM's reasoning process for selecting $[y']$ over the other options. Provide concise explanations for why each reason-element supports $[y']$ as the predicted choice. Focus on the LM's behavior and the significance of the Ranked Reason-elements. Your response should be short and concise.

Explanation (Stage 2): Based on the $[E_{why}]$, explain why this LM makes the other options less likely $[A \setminus \{y'\}]$. Your response should be short and concise.

D Instance Example

We present explanation examples of correct and incorrect predictions in Box.1 and Box.2. `XplainLLM`provides users with a comprehensive understanding of model's reasoning behavior. Additionally, the *debugger-scores* are specifically designed to evaluate the quality of explanations, providing a deeper understanding of the model's behavior from a debugging perspective.

We provide the data schema of our dataset to illustrate the structure of each instance. The schema is outlined below:

Data Schema

question: typeof(string)
answers: typeof(list_of_strings)
label: typeof(string)
predicted_label: typeof(string)
label_matched: typeof(boolean)
concept: typeof(list_of_strings)
topk: typeof(list_of_strings)
explanation_why: typeof(string)
explanation_why_not: typeof(string)
debugger_score: typeof(string)
embedding: typeof(list_of_floats)

E Details of Debugger-Score

The *debugger-score* is a metric that quantifies the quality of the explanations generated by the LLMs. The score evaluates explanations based on multiple dimensions such as faithfulness, accuracy, and completeness. By measuring how well the explanations align with a "perfect" targeted LLM, the debugger score provides a comprehensive evaluation of the generated explanations. This metrics is useful for ensuring that the explanations are not only plausible but also grounded in facts, enhancing trust of explanations generated by LLMs. This instruction assesses explanations based on three dimensions: faithfulness, completeness, and accuracy.

E.1 Instructions for Debugger-score Calculation

Prompt System: Evaluators, assuming the role of LM debuggers with expertise in model parameter changes, assess explanations from the perspective of how model parameters influence decision-making. The assessment focuses on whether the explanation accurately reflects the computational and statistical mechanisms utilized by the LM.

Prompt Content: Evaluators are presented with a task where the LM is augmented with key reasoning elements derived from its operation. This includes the question, answer options, the LM's prediction, and the corresponding explanation.

Evaluation Criteria:

- **Faithfulness:** Does the explanation accurately represent the underlying computational processes and data-driven mechanisms used by the LM to reach its conclusion?
- **Completeness:** Does the explanation encompass all significant computational strategies and data insights relied upon by the LM to make the decision?
- **Accuracy:** How precisely does the explanation reflect the true capabilities and decision-making processes of the LM, considering its design, training data, and functional algorithms?

Scoring: Evaluators are instructed to score each dimension on a scale from 1 to 5, where 1 indicates the lowest level of adherence (poor) and 5 indicates the highest (excellent). The scoring guide emphasizes balanced evaluation, advising against overly strict judgments.

Explanation example of correct prediction

Question: *John cared for Lucy but had trouble expressing it. Lucy was disturbed by John's inability to express affection and felt that he was what?*
Answers: A. Being mean, B. Negligence, C. Disinterest, D. Misunderstood, E. Unfeeling
Label: Unfeeling
Predicted Label Index: 4
Label Matched: true
Reason-elements: "enraged", "delay", "abiogenesis", "sneerer", "helpable", "begrudge", "mollify"...
Top-ranked Reason-elements: "enraged", "delay", "abiogenesis", "sneerer", "helpable"
Explanation (Why-choose): The model selected "unfeeling" primarily due to how it processed the emotional descriptors in the scenario and its understanding of the term definitions. Though the key reason-elements like "enraged", "delay", "abiogenesis", "sneerer", and "helpable" seem unrelated at first, they could connect to various aspects of emotional disconnect and misunderstanding.

- "Enraged" hints at strong emotional responses, albeit negative, which possibly led the model to associate less expressed emotions to being "unfeeling".
- The term "sneerer", indicating derision, might be linked by the model to John's perceived dispassion.
- The word "helpable" suggests a situation needing improvement, reflecting on Lucy's feelings towards John's emotional expression deficits.

Explanation (Why-not-choose):

- **"being mean":** The context didn't explicitly describe John's behavior as intentionally harmful or spiteful, which "being mean" would imply. The reason-elements don't directly associate with malicious intent.
- **"negligence":** Although John's behavior could be interpreted as neglectful, the elements like "enraged" do not support an outright disregard, but rather an emotional complexity.
- **"disinterest":** This choice could somewhat fit, but the model likely found stronger connections in emotional response terms suggesting not just a lack of interest but a lack of feeling.
- **"misunderstood":** While the context and reason-elements like "sneerer" might suggest misunderstandings, 'unfeeling' directly refers to a perceived absence of emotion which seemed more fitting to Lucy's disturbance.

Debugger-score: Faithfulness: 4/5 | Completeness: 3/5 | Accuracy: 4/5

Box 2: Explanation example of correct prediction.

E.2 Faithfulness Evaluation

We explore the faithfulness of explanations split by the correctness of the LLM's predictions. The results in Table 6 show that explanations for correct predictions tend to score higher in faithfulness, completeness, and accuracy compared to those for incorrect predictions, indicating that explanations for incorrect answers often reflect less reasonable reasoning.

Prediction Type	Faithfulness	Completeness	Accuracy	Overall
Correct Prediction	0.74	0.66	0.77	0.72
Incorrect Prediction	0.66	0.59	0.68	0.64

Table 6: Debugger-Scores Split by Prediction Correctness. Explanations for correct predictions score higher, while incorrect predictions reflect less reasonable reasoning.

Explanation example of incorrect prediction

Question: *What do parents tell a child to do on the weekend?*
Answers: A. Study, B. Begin school, C. Go out to play, D. Row boat, E. Clean room
Label: Go out to play
Predicted Label Index: 4
Label Matched: false
Reason-elements: "place_baseball", "asleep", "subway", "ups", "mentors", "renuka", "movable_estate"...
Top-ranked reason-elements: "place_baseball", "asleep", "subway", "ups", "mentors"
Explanation (Why-choose): The model's prediction of "E. Clean room" as the task parents tell a child to do on the weekend is influenced by the ranked reasoning elements, each contributing to the rationale in abstract but supportive ways:

1. **Place Baseball:** Suggests activities requiring cleanup, reinforcing the idea of tidying up spaces like a room.

2. **Asleep:** Implies end-of-day tasks, where cleaning up could be a preparatory action before rest.

3. **Subway:** Though less direct, could symbolize daily routines, pointing to regular tasks like room cleaning.

4. **Ups:** Might relate to picking up or organizing items, aligning with cleaning activities.

5. **Mentors:** Reflects the guiding role of parents in instilling responsibility, including maintaining cleanliness.

Together, these elements hint at themes of responsibility, routine, and organization, leading to the conclusion that cleaning a room is a fitting weekend task for a child from a parent's perspective.
Explanation (Why-not-choose):

- **A. Study and B. Begin school:** They're associated with cognitive tasks, diverging from the model's focus on physical organization and day-end routines.

- **C. Go out to play:** Suggests initiating a new activity, not related to the model's emphasis on tidiness and concluding the day with responsible actions.

- **D. Row boat:** Similar to "C," it indicates starting an activity rather than organizing or cleaning, moving away from the themes of responsibility towards personal spaces.

Debugger-score: Faithfulness: 1/5 | Completeness: 2/5 | Accuracy: 1/5

Box 3: Explanation example of incorrect prediction.

Examples of correct and incorrect predictions, and their explanations, can be found in Appendix D.

F Experiments

In this section, we describe the details of our evaluation that are omitted in Section 5 due to space constraints.

F.1 Model Parameters

To train our GNN, we use a dropout rate of 0.2, a batch size of 64, and a learning rate of 1e-5, optimized with RAdam. The model is fine-tuned on a single NVIDIA A100 GPU for approximately 3 hours. Our KG containing 799,273 nodes and 2,487,810 edges. Our g_e is pruned based on KG to retain 200 high-ranking nodes with a hop size of 2. The GNN, specifically, consists of 200 dimensions

Model	Shots	Version	Faithfulness	Completeness	Accuracy	Overall
gpt-3.5-turbo	Five-shot	Vanilla	**0.70**	0.59	**0.73**	**0.67**
		XplainLLM	0.69	**0.61**	0.71	0.67
	Two-shot	Vanilla	0.70	0.59	0.73	0.67
		XplainLLM	**0.71**	**0.67**	**0.73**	**0.70**
llama3-8b	Five-shot	Vanilla	**0.70**	0.53	**0.72**	**0.65**
		XplainLLM	0.67	**0.57**	0.67	0.64
	Two-shot	Vanilla	0.70	0.53	0.72	0.65
		XplainLLM	**0.71**	**0.71**	**0.79**	**0.74**

Table 7: Results of Vanilla and `XplainLLM` Versions of LLMs with Five-shot and Two-shot.

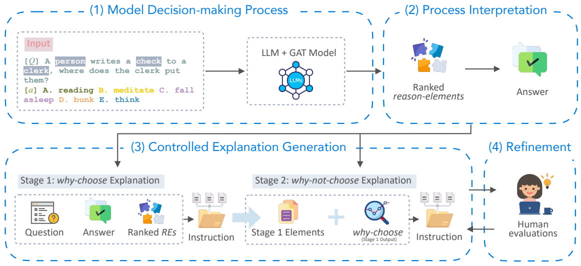

Figure 8: Data Collection Process.

and 5 layers. The learning rate in our experiments is 1e-3.

For Llama-3-8B and RoBERTa-large models, the learning rate is set to 1e-5 and a batch size of 8, optimizing with AdamW. We used early stopping based on validation loss to prevent overfitting.

F.2 Sensitivity to In-Context Example Size

As discussed in our results, GPT-3.5-turbo and Llama-3-8B display suboptimal performance in generating explanations under certain conditions. This can be attributed to the inherent limitations of these models when it comes to generalizing in-context examples. Their architectures may restrict their ability to fully leverage the context provided or effectively integrate external knowledge during the inference phase.

In our primary experiments, we use a five-shot setting to ensure fair comparisons across different models. We present the results for GPT-3.5-turbo and Llama-3-8B under two-shot and five-shot conditions in Table 7.

The results from the two-shot experiment show that both GPT-3.5-turbo and Llama-3-8B have improved performance in the two-shot setting compared to the five-shot configuration. In particular, the results suggest that these models may struggle to generalize from a larger number of in-context examples (long-context), and perform better when presented with fewer, more targeted examples.

G Detailed Data Collection

Figure 8 shows the process of data collection:

1. Given a question, we retrieve its relevant knowledge using the KG. The retrieved graph is then pruned based on scores influenced by the LLM, resulting in what we term the *element-graph*. The *element-graph* is processed by a specialized GAT model (known formally as Decision Interpretation). Leveraging attention mechanisms, we obtain the essential representations for interpretation.

2. The model's decision-making is interpreted through the ranked *reason-elements* and the predicted answer.

3. A controllable dual-stage process generates the explanations: Stage 1: The initial phase focuses on generating the "why-choose" explanations. Stage 2: Building upon the outputs and elements of Stage 1, we then generate the "why-not-choose" explanations.

4. We conduct a human evaluation to identify errors in the explanations. If discrepancies arise, explanations are reverted to Step 3 for refinement. This process not only helps prevent potential issues arising from bad explanations generated by the LLM but also maintains human-aligned clarity and relevance.

H Impact of `XplainLLM` Framework on Explanation Quality

H.1 Comparison of Explanations: Vanilla vs. `XplainLLM`

We compare the explanations generated by the vanilla model (GPT-4-turbo) and the model enhanced with the `XplainLLM` framework below. This query involves choosing the correct venue where tickets for games are typically sold, highlighting how `XplainLLM` grounds the reasoning in real-world knowledge.

Question: *Where do you buy tickets at a ticket booth for games?*

Answer Options: A. train station, B. cathedral, C. metro station, D. fairgrounds, E. amusement park

Ground Truth: **D. fairgrounds**

Vanilla Explanation (Prediction: E. amusement park - Incorrect):

Among the options provided, "amusement park" is the most plausible location where you would find ticket booths specifically for games, as amusement parks typically feature various games and attractions that require tickets. Other options like train stations, metro stations, and cathedrals generally do not involve game-related activities, and while fairgrounds might also offer games, they are not as directly associated with ticket purchases for games as amusement parks.

`XplainLLM` Explanation (Prediction: D. fairgrounds - Correct):

The model selects "fairgrounds" as the correct answer. This decision is grounded in the model's understanding of key vocabulary and context, supported by the following reasoning elements:

- **entertainment_venues**: Fairgrounds are typical examples of venues where various games and entertainment activities are held, often requiring ticket booths for access.
- **public_gathering_spots**: Fairgrounds are associated with large public gatherings, particularly during events or fairs, making them plausible locations for ticket booths.
- **recreational_area**: This term refers to areas designed for leisure and amusement, aligning well with the context of fairgrounds.
- **ticket_purchases**: This phrase is directly applicable to the context of buying tickets, further reinforcing fairgrounds as the appropriate choice.
- **game_access**: Fairgrounds often host multiple games and attractions, where tickets are required for entry, making it the most suitable option.

These elements collectively guide the model towards selecting "fairgrounds" as the most relevant and contextually appropriate option, demonstrating how `XplainLLM` improves the faithfulness and grounding of explanations compared to the vanilla model.

H.2 Impact of KGs on Explanation Quality

The construction of the `XplainLLM` dataset utilizes KGs to provide more grounded and accurate explanations. The quality of the explanations is related to the relevance and reliability of the KGs used. In this study, we employ ConceptNet, which is well-suited for commonsense reasoning tasks due to its extensive coverage of general knowledge, including more than 13 million links between concepts.

However, the effectiveness of `XplainLLM` may vary for domain-specific tasks, such as those in medical or legal fields. In such cases, domain-specific KGs would be needed to ensure optimal performance.

I Explanation Statistics

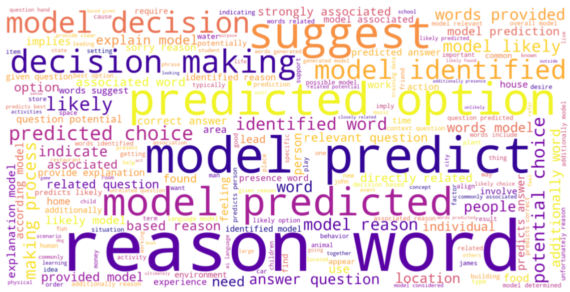

Figure 9: *why-choose* explanations.

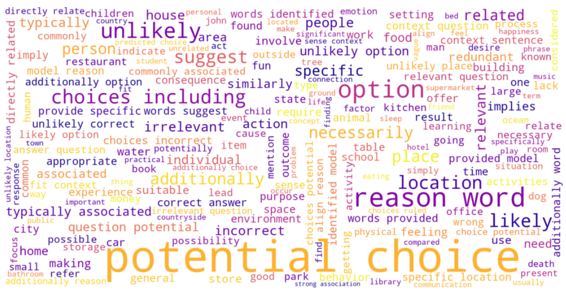

Figure 10: *why-not-choose* explanations.

Figure 9 is a word cloud showing the most frequently appearing words in the *why-choose* explanations. From this figure, we have a clear indication that *why-choose* explanations focus on explaining, comprehension, and interpreting predictions made by the target model.

Figure 10 presents a word cloud for *why-not-choose* explanations. We note that these explanations outline the reasons behind the non-selection of specific options as predicted answers. Furthermore, *why-not-choose* explanations emphasize how the target model determines the likelihood of different answer choices. We also observe that the target model handles a wide array of topics, which can be crucial components in the "why not" reasoning process.

J Evaluation Materials

J.1 Questions and Evaluation Instructions

For each instance, we include a set of question, answer choices, model prediction, and explanation. To evaluate the quality of the explanation, we provide seven questions for evaluators. Each question includes three score levels: 1 for disagree, 2 for neutral, and 3 for agree. The questions and instructions in our evaluation are as follows:

Q0: This is a good explanation

1. **Disagree**: The explanation is illogical or inconsistent with the question and/or does not adequately cover the answer choices.

2. **Neutral**: The explanation is somewhat logical and consistent with the question but might miss some aspects of the answer choices.

3. **Agree**: The explanation is logical, consistent with the question, and adequately covers the answer choices.

Q1: I understand this explanation of how the AI model works.

1. **Disagree**: The explanation is unclear or contains overly complex terms or convoluted sentences.

2. **Neutral**: The explanation is somewhat understandable but might contain complex terms or convoluted sentences.

3. **Agree**: The explanation is clear, concise, and easy to understand.

Q2: I trust this explanation of how the AI model works.

1. **Disagree**: The explanation is unclear or contains overly complex terms or convoluted sentences.

2. **Neutral**: The explanation is somewhat credible but contains some elements that I find doubtful or questionable.

3. **Agree**: The explanation is credible and aligns with my understanding of how AI models work.

Q3: This explanation of how the AI model works is satisfying.

1. **Disagree**: The explanation does not meet my expectations and leaves many questions unanswered.

2. **Neutral**: The explanation somewhat meets my expectations but leaves some questions unanswered.

3. **Agree**: The explanation meets my expectations and satisfies my query.

Q4: This explanation of how the AI model works has sufficient detail.

1. **Disagree**: The explanation lacks detail and does not adequately cover the AI model's decision-making.

2. **Neutral**: The explanation provides some detail but lacks thoroughness in covering the AI model's decision-making.

3. **Agree**: The explanation is thorough and covers all aspects of the AI model's decision-making.

Q5: This explanation of how the AI model works seems complete.

1. **Disagree**: The explanation does not adequately cover the answer choices and leaves many aspects unexplained.

2. **Neutral**: The explanation covers most answer choices but leaves some aspects unexplained.

3. **Agree**: The explanation covers all answer choices and leaves no aspect unexplained.

Q6: This explanation of how the AI model works is accurate.

1. **Disagree**: The explanation does not accurately reflect the AI model's decision-making.

2. **Neutral**: The explanation somewhat reflects the AI model's decision-making but contains some inaccuracies.

3. **Agree**: The explanation accurately reflects the AI model's decision-making.

J.2 Human-centered Metrics for Explanation Quality Evaluation

The meaning of metrics used in the human-centered evaluation are as follows:

1. **Overall quality** reflects the overall effectiveness of explainability. It reveals how effectively explanations convey the decision-making process of the AI models to the human users.

2. **Understandability** evaluates how well a human can comprehend the model's output and explanations. It captures the clarity and coherence of the generated text.

3. **Trustworthiness** measures the human evaluator's confidence in the model's outputs and explanations. It evaluates whether the explanations appear reliable, credible, and based on sound reasoning.

4. **Satisfaction** captures the overall contentment of the evaluator with the explanations. It measures whether the outputs meet the evaluator's needs and expectations in terms of quality, relevance, and utility.

5. **Sufficiency of detail** evaluates whether the explanations provide a sufficient level of detail. It evaluates whether the responses are adequately descriptive and provide all necessary information to fully answer the question or task.

6. **Completeness** measures whether the explanations address the decision behaviors of the model.

7. While we also measure **accuracy** objectively, the human evaluation of accuracy assesses whether the explanations align with the evaluator's knowledge or expectations. It measures whether the explanations can reflect if the model's outputs are factually correct and contextually appropriate.