Reasoning-Grounded Natural Language Explanations for Language Models

Vojtech Cahlik, Rodrigo Alves, and Pavel Kordik

Faculty of Information Technology, CTU in Prague, Prague, Czech Republic
{vojtech.cahlik,rodrigo.alves,pavel.kordik}@fit.cvut.cz

Abstract. We propose a large language model explainability technique for obtaining faithful natural language explanations by grounding the explanations in a reasoning process. When converted to a sequence of tokens, the outputs of the reasoning process can become part of the model context and later be decoded to natural language as the model produces either the final answer or the explanation. To improve the faithfulness of the explanations, we propose to use a joint predict-explain approach, in which the answers and explanations are inferred directly from the reasoning sequence, without the explanations being dependent on the answers and vice versa. We demonstrate the plausibility of the proposed technique by achieving a high alignment between answers and explanations in several problem domains, observing that language models often simply copy the partial decisions from the reasoning sequence into the final answers or explanations. Furthermore, we show that the proposed use of reasoning can also improve the quality of the answers.

Keywords: Explainable AI · Large Language Models · Natural Language Explanations · Reasoning

1 Introduction

Today's prevalent large language model (LLM) explainability techniques lack the expressivity of natural language, as the explanations are limited in detail and hard to interpret for an untrained user [1]. On the other hand, natural language explanations [2] can potentially be easy to follow and unlimited in expressivity, but their faithfulness is typically questionable, such as with the simple *answer-then-explain* setting which tends to lead models into fabulating their explanations. Moreover, it is even questionable whether LLMs produce their outputs in a thought process that is anyhow related to human reasoning, as they are in essence mere enhancements of traditional n-gram models [3]. Chain-of-thought reasoning is one notable improvement of the decision process as the answers tend to follow from the preceding natural language reasoning sequences, but it is too computationally intensive for ubiquitous use.

We propose to ground natural language explanations, as well as the answers, in a suitable resource-efficient LLM reasoning process. When converted to a sequence of tokens, the result of the reasoning process can then become part of

the context observed by the model when producing its final answer or explanation. The reasoning sequence does not have to be directly human-readable, as it merely has to encode the explanation together with the answer. This information can then be simply decoded from the reasoning sequence to natural language when the model generates the final answer or explanation. In order for the explanations to be credible, a joint predict-explain setting can be used, in which the answer and explanation are inferred independently of each other. To demonstrate the plausibility of this approach, we experiment with compact reasoning sequences that we refer to as *compressed chain-of-thought reasoning*. We present the high-level overview of our methodology in Figure 1.

We evaluate our explainability framework in an "LLM-as-a-classifier" setting, in which we train LLMs to mimic the behavior of simple machine learning classifiers such as decision trees. This setting is convenient for our approach as it can be framed so that the final model outputs are affected by multiple intermediate decisions, and also allows for simple and deterministic evaluation of results. Our paper makes the following contributions to the field of LLM explainability:

1. We propose an LLM explainability technique for producing faithful natural language explanations grounded in reasoning.

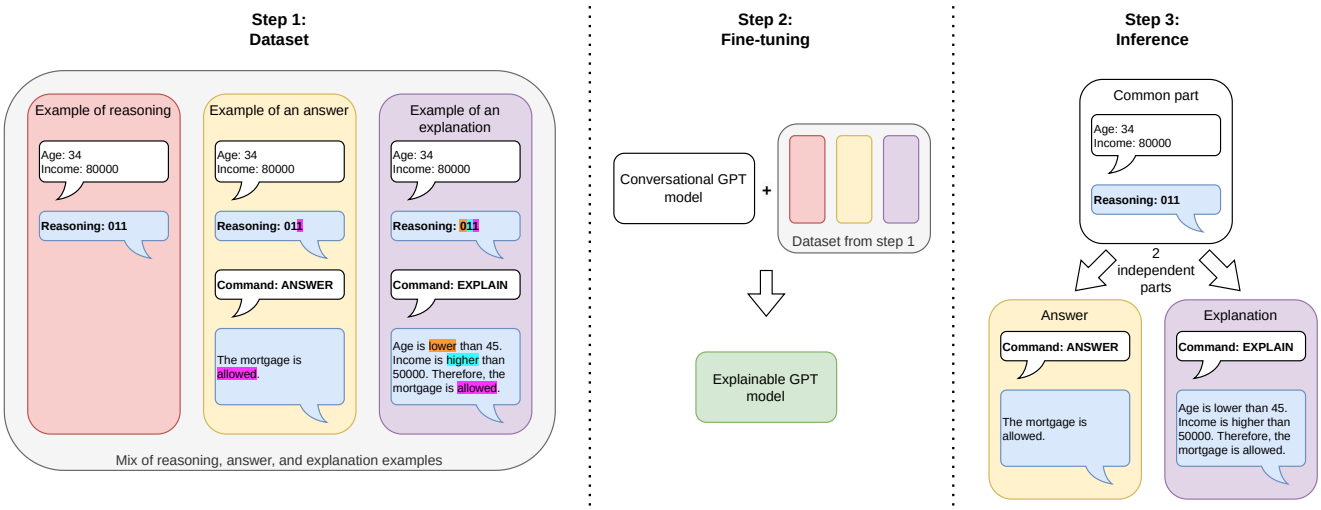


Fig. 1: Overview of our methodology. As a first step, we gather a conversational dataset in which for each user input, the triplet of reasoning-answer-explanation ground truths is present. In the second step, we fine-tune a conversational GPT model on the dataset from step 1. As a last step, we perform inference using the fine-tuned model by first computing a reasoning sequence and then including it in the conversation to produce the final answer or explanation, which are obtained independently of each other.

2. We observe that when a suitable reasoning process is included in LLM training, and the outputs of the reasoning process are placed in the LLM input contexts, LLMs will often copy the partial decisions from the reasoning sequence into their answers or explanations.
3. We demonstrate the plausibility of our proposed explainability technique by achieving a high alignment between answers and explanations in several problem domains.
4. We show that besides enabling faithful natural language explanations, the inclusion of the reasoning process can also improve the quality of answers.

2 Related Work

2.1 LLM Explanations

LLMs are complex black boxes, and without proper explainability, it is difficult to understand their capabilities, limitations, and potential failures. [1,4]. Explainability techniques are commonly categorized according to several criteria: whether explainability is incorporated into the model's architecture and thus the explanation is part of the model's prediction (*ante-hoc* or *intrinsic explanations*) or if explanations are calculated after the model has been trained and a prediction has been obtained (*post-hoc explanations*); or whether they are related to a single prediction (*local explanations*) or to the general behavior of the model across all predictions (*global explanations*).

[1] categorizes local LLM explainability techniques into 4 main approaches: feature attribution-based explanations, attention-based explanations, example-based explanations, and natural language explanations.

Feature Attribution-based Explanations These explanations measure the importance of each input feature (such as an input token) in relation to outputs. *Perturbation-based techniques* perturb the inputs using removal or masking [5,6], which may however generate out-of-distribution data. *Gradient-based* techniques measure partial derivatives of outputs with respect to the input features, using well-established explainability techniques such as *gradient* $\times$ *input* or *integrated gradients* [7,8,9,10] which address some of the difficulties that occur when using gradients naively [11]. *Surrogate model methods* employ simpler white-box models to explain individual predictions, notably using the SHAP technique, which utilizes Shapley values and has also been adapted to Transformer models [12].

Attention-based Explanations Attention-based explanations analyze the parameters or behavior of attention heads. Numerous studies have focused on explaining attention heads using visualizations, such as with token-level bipartite graphs and heatmaps or neuron-level heatmaps [13,14]. Other works have adopted gradient-based methods using various definitions of gradient in attention heads [10,15]. However, there is ongoing debate on the reliability of attention-based explainability techniques [1].

Example-based Explanations These explanations analyze how changes in model inputs affect the outputs. A popular approach is to generate *counterfactual examples* that cause important changes in the outputs by adding, altering, masking, removing, or shuffling words in the input text [16]. On the other hand, *adversarial examples* aim to substantially alter the model outputs with barely noticeable changes to the input text [17,18]. These examples can be added to the training data to improve the robustness of the final model. Another family of methods aims to analyze the impact of the individual training examples on the behavior of the trained model, remarkably without the need for multiple rounds of training [19,20].

Natural Language Explanations Natural language explanations refer to explanations that take the form of text in natural language, thus making them suitable even for a lay audience [1,2]. The quality of natural language explanations is commonly assessed according to plausibility, which checks if the explanations are logically sound, faithfulness, which assesses whether the explanations describe the true decision process of the model, and readability [21]. Although being a relatively large field, most natural language explanation studies focus on other types of models than LLMs. The approaches for LLMs include using simple explain-then-predict and predict-then-answer methods [22], training the models using datasets of synthetic [23] or human-written explanations [24,25], and translation of natural language to symbolic solver domains [26].

2.2 LLM Reasoning

The field of LLM reasoning covers a wide range of methods aimed at improving the model outputs or answers. Chain-of-Thought [27] is perhaps the most well-known technique, in which the LLM is simply tasked to reason first before stating the final answer. Extensions of this approach include self-consistency [28], in which multiple reasoning paths are sampled, and Tree of Thoughts [29], where the reasoning trajectories form a tree which is explored using search strategies such as BFS and DFS. Other notable reasoning methods include multi-agent collaboration [30], knowledge distillation [31], process-based reward models [32], Monte Carlo Tree Search [33], and reinforcement learning [34].

3 Methods

3.1 Reasoning-Grounded Natural Language Explanations

To achieve faithful natural language explainability, we propose to ground LLM explanations as well as answers in a reasoning process. In order to decrease computational complexity, we suggest that the output of the reasoning process does not have to be inherently human-readable, but that it should merely contain the information necessary to be later decoded by the LLM into the final answer or

natural language explanation. Such reasoning can be used in a two-step conversational framework, where as the first step, the reasoning sequence is generated, and as the second step, the user (or the chat interface) sends a *command* message indicating whether the model should answer the question or explain the answer, and the model responds accordingly. In case the user chooses to obtain both the answer as well as the explanation, it is crucial that both are obtained independently by the chat interface to prevent the model from fabulating the explanation or the answer being affected by the explanation.

For a clearer definition of the conversational inference process, we can define the conversation history H_n as a sequence of user question messages U_i and model answer messages A_i:

$$H_n = U_1 \cdot A_1 \cdot U_2 \cdot A_2 \cdot \ldots \cdot U_{n-1} \cdot A_{n-1} \cdot U_n \tag{1}$$

With this notation, the message with reasoning output R_n can be defined by the formula

$$R_n = \text{ReasoningModel}(H_n), \tag{2}$$

and the model's answer message A_n and explanation message E_n can be defined by the formulas

$$A_n = \text{LLM}(H_n \cdot R_n \cdot C_{\text{answer}}) \tag{3}$$

and

$$E_n = \text{LLM}(H_n \cdot R_n \cdot C_{\text{explain}}), \tag{4}$$

where C_{answer} and C_{explain} are the "ANSWER" and "EXPLAIN" command messages.

As a proof of concept, we experiment with using the LLM to generate compact reasoning sequences in an approach that we refer to as *compressed chain-of-thought reasoning*. With the use of the previous notation, we therefore set ReasoningModel = LLM. We put three requirements on our compressed chain-of-thought reasoning sequences:

1. The reasoning sequences should encode all the partial decisions necessary for the model to produce the right answers.
2. The reasoning sequences should encode all the partial decisions necessary for the model to produce the natural language explanations in the desired level of detail.
3. The encoding of the partial decisions in the reasoning sequences should follow a chain-of-thought ordering to allow accurate token-by-token generation by the LLM.

Similarly to regular chain-of-thought reasoning, compressed chain-of-thought reasoning can also potentially improve the quality of LLM answers, as more circuit layer operations can be performed by the LLM before the final answer is produced.

Table 1: Examples of instances from our experimental datasets. For each instance, the sections corresponding to values included in the reasoning sequence are highlighted with bold text, and the occurrences of the ground truth class are underlined.

	Dataset		
	Logistic regressor	**Decision tree**	**Natural language decision tree**
Input	X: [-0.4408, 0.7812, -0.3482, 0.9094, 0.869, -0.0214, -0.0555, -0.8395]	X: [0.923, 0.252]	Loan amount: $115000.0 Loan-to-value ratio: 92.266 Debt-to-income ratio: <20% Applicant's age: 25-34 Loan term: 120.0 Income: $83000.0 Property value: $475000.0 Total loan costs: $0.0
Reasoning	**-1.2465 -1.2465;-2.9536 -4.2001;-2.3885 -6.5886;7.2595 0.6709;-4.5762 -3.9053;0.2138 -3.6915;-0.5065 -4.198;6.8913 2.6933;1**	**0,0,0,1,1,1,0,0**	**1,0,0,0,1**
Answer	**1**	**0**	The mortgage is **issued**.
Explanation	[[0, "w[0] * x[0] = **-1.2465**", "y - 1.2465 = **-1.2465**"], [1, "w[1] * x[1] = **-2.9536**", "y - 2.9536 = **-4.2001**"], [2, "w[2] * x[2] = **-2.3885**", "y - 2.3885 = **-6.5886**"], [3, "w[3] * x[3] = **7.2595**", "y + 7.2595 = **0.6709**"], [4, "w[4] * x[4] = **-4.5762**", "y - 4.5762 = **-3.9053**"], [5, "w[5] * x[5] = **0.2138**", "y + 0.2138 = **-3.6915**"], [6, "w[6] * x[6] = **-0.5065**", "y - 0.5065 = **-4.198**"], [7, "w[7] * x[7] = **6.8913**", "y + 6.8913 = **2.6933**"], ["OUTPUT", **1**]]	[["0.923 < 0.3562", **false**], ["0.252 > 0.6825", **false**], ["0.923 < 0.5613", **false**], ["0.252 < 0.2597", **true**], ["0.923 > 0.8087", **true**], ["0.252 > 0.0709", **true**], ["0.923 < 0.8676", **false**], ["OUTPUT", **0**]]	The loan-to-value ratio is **higher than** 79%. The income is **lower than** $110000. The applicant's age is **lower or equal to** 34 years. The debt-to-income ratio is **lower or equal to** 40%. Therefore, the mortgage is **issued**.

3.2 LLM as a Classifier

In our experiments, we adopt an "LLM-as-a-classifier" approach in which the LLM is tasked with mimicking the behavior of a machine learning classifier. This approach is convenient for our study as reasoning and explanation sequences can be formulated so that they involve chains of various decisions, and we can calculate evaluation metrics deterministically without using methods such as LLM-as-a-judge. We experiment with three problem domains: Logistic regression, decision tree classification, and a natural language dataset generated using decision tree logic. For each problem domain, we design the ground-truth answers to simply state the classification result. For explanations, we use detailed chain-of-thought sequences that describe the intermediate decisions necessary to reach correct classification, and for reasoning, we extract or encode the most important values from the explanation sequence to form minimal, "compressed" chain-of-thought sequences. For each of the datasets, an example of an input, reasoning, answer, and explanation text for a single instance is shown in Table 1.

Logistic Regression In this setting, we randomly generate a parameter vector $\mathbf{w}$ of a logistic regression model without a bias parameter and train the LLM to classify random 8-dimensional input vectors $\mathbf{x}$ according to the following formula:

$$y(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w}^T\mathbf{x} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Decision Tree In this setting, we randomly generate a binary decision tree of depth 7 with the following node selection logic at each non-leaf node:

$$next(N) = \begin{cases} left(N) & \text{if } s_N \times x_{index(N)} > s_N \times t_N \\ right(N) & \text{otherwise,} \end{cases} \tag{6}$$

where $next(N)$ is the next node to be evaluated after the current node N, $left(N)$ and $right(N)$ are the left and right child nodes of node N, t_N is a random threshold, $index(N)$ is a function that selects the index of $\mathbf{x}$ (defined so that two consecutive nodes can not use the same index), and s_N is a random sign of -1 or 1 that can effectively flip the comparison operator.

Leaf nodes are assigned a class of 0 or 1.

Decision Tree Encoded in Natural Language In the last setting, we experiment with a decision tree that represents a mortgage application review process encoded in natural language. We take a subset of randomly selected mortgage applications from the 2022 version of the *HMDA National Loan Level Dataset* [35] as input data and using a manually designed decision tree that represents a fictional mortgage application review process, we generate paragraphs in which each sentence describes a decision branch comparison for one of the input features. Decisions are evaluated for each dataset instance from the top of the decision tree to the leaf with the final class of issued or not issued.

4 Experiments

4.1 Categorization of Experiments

Separate Fine-Tuning for Answers and Explanations In this setting, the training dataset is split into two separate datasets, each composed of either input-command-answer or input-command-explanation instances. The LLM is then fine-tuned on each of the two datasets independently, resulting in two fine-tuned models. The inference for answers and explanations is then performed separately using the corresponding model.

Joint Fine-tuning This setting is similar to the previous one, but the answer and explanation instances are not separated into two training datasets. Instead, fine-tuning is performed jointly on the mix of input-command-answer and input-command-explanation examples. Inference is performed in the joint predict-explain approach, with the answers generated independently of the explanations and vice versa, according to the command "ANSWER" or "EXPLAIN".

Joint Fine-tuning with Reasoning In this setting, the training dataset is composed of a mix of input-reasoning, input-reasoning-command-answer, and input-reasoning-command-explanation instances. Inference is performed in two steps, where in the first step, the model generates a reasoning sequence, and in the second step, the answer and explanation are generated independently according to the "ANSWER" or "EXPLAIN" command.

In-context Learning To understand how strongly the performance of LLMs is affected by fine-tuning to the specific problem domain, we include results for in-context learning [36] as an informative baseline. In this setting, the LLMs are not fine-tuned to the specific classification model, but instead obtain their problem domain knowledge only from classification input-output example pairs included in their input prompt. For each few-shot example, both the answer and explanation target is included. In order to prevent the influence of human prompt engineering, we pre-train the LLMs on a training dataset where each training instance belongs to a different problem domain, corresponding to a randomly generated classifier from the same model family but with different values of model parameters than those used in the test dataset. The input of each few-shot example is randomly generated to achieve greater diversity. We omit the in-context learning setting in the experiments with natural language decision trees due to the complexity of random generation of meaningful decision processes in this problem domain.

4.2 Experimental Setup

All experiments were performed using a similar methodology.[1] For each of the five LLMs tested, the model's instruction-tuned variant was used. LLMs were trained using low-rank adaptation [37] and Adam optimizer [38] for a single epoch on a train dataset created using 2000 classification inputs. During training, the test loss was periodically measured on a test dataset created using 200 inputs, and at the end of training, the best model checkpoint was kept. In the in-context learning experiments, the number of few-shot examples was 5 for logistic regression and 20 for decision trees. The same training hyperparameters were used in all experiments, namely a batch size of 4 and a learning rate of 5×10^{-5} with a linear schedule and 100 warmup steps.

[1] The source code is available under the MIT license at `https://github.com/vcahlik/reasoning-grounded-explanations`, together with our datasets.

4.3 Evaluation Metrics

In our experiments, we separately measure the classification accuracy of answers and explanations. For explanations, determining the resulting classification is possible as we have designed the explanations as chain-of-thought sequences in which the output class is always stated at the end. Furthermore, we measure the rate of alignment between the answer and explanation classifications.

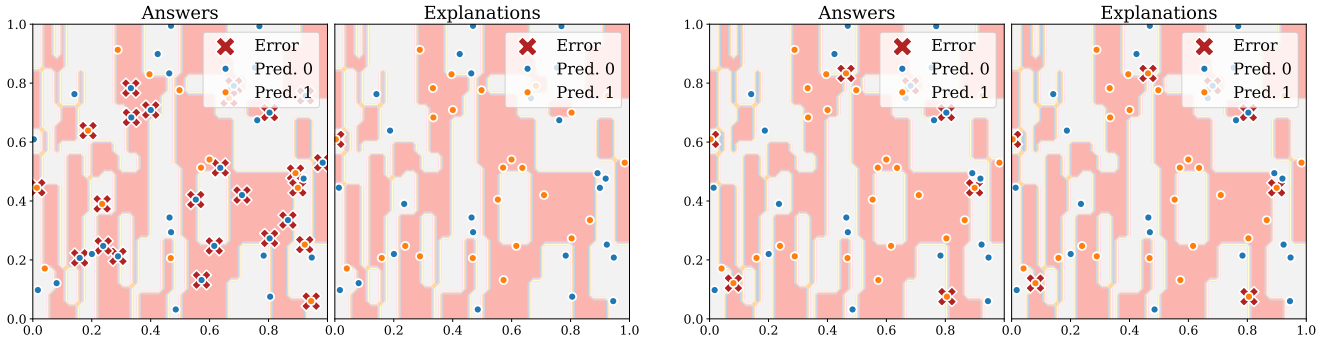
5 Results

5.1 Logistic Regression Results

The results for the logistic regression dataset are shown in Table 2. In-context learning has near-perfect classification accuracy for explanations, as the parameters of the logistic regressor are stated in the few-shot examples and therefore it is simple for the model to generate correct chain-of-thought explanations. However, for answers, classification accuracy is equivalent to random guessing due to the difficulty of the task when a chain-of-thought process is not used. The gap between the classification accuracy for answers and explanations is also wide for most of the fine-tuning experiments without reasoning. However, when reasoning is used, the classification accuracies of answers increase to the level of classification accuracies for explanations, indicating that the reasoning process helps the LLMs achieve correct answers.

Table 2: Results on the logistic regression dataset. The experimental setup differs in whether training was performed separately or jointly for answers and explanations, whether in-context learning (ICL) was used, and whether reasoning was used. Outputs that could not be parsed into a valid class are counted towards errors and their rate is additionally shown in italic.

Ans./exp. training	**ICL**	**Reasoning**	**Metric**	**Llama 3 8B**	**Mistral NeMo**	**Mistral 7B Zephyr SFT**		**Phi-4**
Separately	Yes	No	Answer acc.	0.455	0.515	0.470	0.495	0.470
			Explanation acc.	0.990	1.000	0.995	1.000	0.990
			Alignment rate	0.455	0.515	0.475	0.495	0.460
Separately	No	No	Answer acc.	0.610	0.890	0.830	0.555	0.620
			Explanation acc.	0.615 *(0.140)*	0.990 *(0.005)*	0.995	0.995	0.990
			Alignment rate	0.455 *(0.140)*	0.880 *(0.005)*	0.835	0.560	0.620
Jointly	No	No	Answer acc.	0.640	0.530	0.555	0.470	0.470
			Explanation acc.	0.990	1.000	1.000	0.995	1.000
			Alignment rate	0.630	0.530	0.555	0.475	0.470
Jointly	No	Yes	Answer acc.	0.890 *(0.020)*	0.995	1.000	1.000	0.945
			Explanation acc.	0.875 *(0.030)*	0.995	1.000	1.000	0.940 *(0.005)*
			Alignment rate	0.965 *(0.035)*	1.000	1.000	1.000	0.995 *(0.005)*



(a) Results without reasoning

(b) Results with reasoning

Fig. 2: Experiments with joint training of answers and explanations on a decision tree dataset. The colored regions correspond to ground-truth classes. When reasoning is used, answer and explanation classification errors are typically near-perfectly aligned.

5.2 Decision Tree Results

The results for the decision tree dataset, belonging to a tree of depth 7, are shown in Table 3. Even though 4 times as many few-shot examples were used than in the logistic regression experiments, the classification accuracy of in-context learning is low even for explanations, as the number of few-shot examples is still lower than the number of decision tree leaves. The results for fine-tuning without reasoning are similar to those with the logistic regression dataset. Results for reasoning show higher error rates than in the logistic regression experiments, presumably due to the less detailed reasoning process. However, answers and explanations now remarkably contain the same classification errors in almost all cases, as can be seen from the near-perfect alignment rates. We visualize this phenomenon by plotting the classifications for one of the experiments in Figure 2.

Figure 3 shows the results for the Mistral 7B model on datasets of varying decision tree depths, indicating that classification accuracy tends to decrease as the complexity of the trees increases. With reasoning, answer and explanation classifications are near-perfectly aligned for all of the depths, in contrast to the alignment rates for experiments without reasoning. However, explanations without reasoning tend to have the highest classification accuracy in these experiments, supposedly due to the chain-of-thought explanation sequences being more thorough than the compressed chain-of-thought reasoning sequences.

5.3 Natural Language Decision Tree Results

The results for the natural language decision tree dataset, shown in Table 4, are similar to those for the decision tree dataset. However, in this case, the

Table 3: Results on the decision tree dataset, with the same semantics as in Table 2

Ans./exp. training	ICL	Reasoning	Metric	Llama 3 8B	Mistral NeMo	Mistral 7B	Zephyr SFT	Phi-4
Separately	Yes	No	Answer acc.	0.535	0.510	0.490	0.490	0.535
			Explanation acc.	0.670	0.685	0.695	0.695	0.700
			Alignment rate	0.565	0.565	0.495	0.485	0.505
Separately	No	No	Answer acc.	0.475	0.525	0.530	0.565	0.565
			Explanation acc.	0.975	0.985	0.985	1.000	0.955
			Alignment rate	0.480	0.540	0.515	0.565	0.580
Jointly	No	No	Answer acc.	0.475	0.450	0.500	0.475	0.520
			Explanation acc.	0.985	0.985	0.995	0.975	0.985
			Alignment rate	0.480	0.435	0.505	0.460	0.505
Jointly	No	Yes	Answer acc.	0.745	0.835	0.845	0.875	0.715
			Explanation acc.	0.745	0.835	0.840 *(0.005)*	0.875	0.715
			Alignment rate	1.000	1.000	0.995 *(0.005)*	1.000	1.000

use of reasoning is associated with near-perfect classification accuracies for both answers and explanations and with perfect alignment of classification errors.

5.4 Analysis of Errors

As a further analysis, we study the partial decisions present in the reasoning and explanation sequences generated on the decision tree dataset by the Llama 3 8B

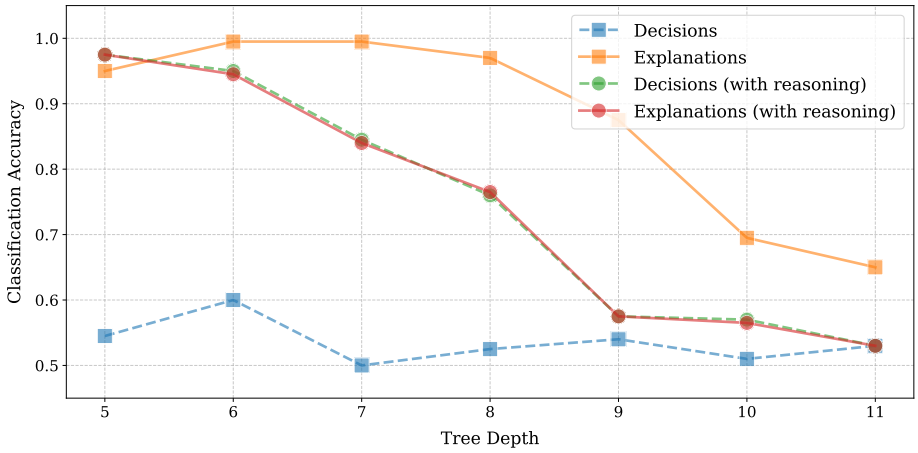


Fig. 3: Classification accuracies for experiments with decision trees of various depths

Table 4: Results on the natural language decision tree dataset, with the same semantics as in Table 2

Ans./exp. training	ICL	Reasoning	Metric	Llama 3 8B	Mistral NeMo	Mistral 7B	Zephyr SFT	Phi-4
Separately	No	No	Answer acc.	0.810	0.825	0.850	0.850	0.815
			Explanation acc.	0.950	0.975	0.950	0.995	0.970
			Alignment rate	0.860	0.840	0.850	0.855	0.835
Jointly	No	No	Answer acc.	0.840	0.840	0.810	0.865	0.845
			Explanation acc.	0.935	0.975	0.980	0.990	0.975
			Alignment rate	0.825	0.855	0.820	0.875	0.830
Jointly	No	Yes	Answer acc.	0.985	0.970	0.985	1.000	1.000
			Explanation acc.	0.985	0.970	0.985	1.000	1.000
			Alignment rate	1.000	1.000	1.000	1.000	1.000

model. As shown in Table 5, the final classification errors in both the reasoning and explanation sequences are caused by the accumulation of mistakes in partial decisions. It is noteworthy that all of the decisions are perfectly aligned between the reasoning and explanation sequences. Although not shown in the table, we also observed perfect alignment between the answer classifications and reasoning classifications, meaning that all partial decisions as well as the final classifications are aligned between the generated answers, explanations, and reasoning in this case.

6 Discussion

It may not be immediately clear why the inclusion of reasoning sequences in LLM input contexts leads to alignment between answers and explanations. It seems that during training, the LLM must learn the relatively simple task of reproducing the compressed chain-of-thought reasoning sequence to succeed at the more difficult task of producing the one-step answer classifications. We hypothesize that once the model learns to produce accurate reasoning sequences, the internal mechanism by which the LLM produces its explanations also degrades to the copying of the partial decisions from the reasoning sequence. To

Table 5: Analysis of the correctness of the reasoning and explanation chain-of-thought sequences generated by Llama 3 8B on the decision tree dataset

	Partial decision							Final classification
	1	2	3	4	5	6	7	
Reasoning accuracy	0.995	1.000	0.990	0.960	0.895	0.830	0.745	0.745
Explanation accuracy	0.995	1.000	0.990	0.960	0.895	0.830	0.745	0.745
Alignment rate	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

gain supporting evidence for this hypothesis, we further experimented with randomly flipping the partial decisions as well as the final classification decisions present in the reasoning sequences produced by fine-tuned Llama 3 8B. As was suspected, we observed that almost all of the changes were propagated into the produced explanations as well as to the answers.

Our approach presented in this paper can be extended in numerous possible ways, which we leave for future work. Primarily, the reasoning process that we chose for our proof-of-concept experiments could be extended to wider problem domains or even to general-purpose assistant datasets, for example by straightforwardly using chain-of-thought reasoning or similar approaches, such as the reasoning process used by DeepSeek-r1 [34]. It would also seem beneficial to introduce a training loss that directly penalizes the mismatch between answers, explanations, and reasoning. Furthermore, we believe that LLM applications could benefit from other output modes besides answering and explaining. We envision a multitask setting with additional implemented commands, such as those for obtaining explanations of varying detail, classification of user intent, content filtering analysis, metadata generation, and so on.

7 Conclusion

In this paper, we have proposed an LLM explainability technique for obtaining faithful natural language explanations by grounding the LLM answers and explanations in a reasoning process. We have shown that LLMs often simply copy the partial decisions from the reasoning sequence into their answers or explanations, and we utilized this phenomenon to achieve high alignment between answers and explanations in several problem domains. Furthermore, we have shown that besides enabling faithful explanations, the use of a reasoning process can also lead to improvements in the quality of answers. We hope that our study inspires further research or real-world use-cases that advance the current state of explainability in LLMs.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38, 2024.
2. Erik Cambria, Lorenzo Malandri, Fabio Mercorio, Mario Mezzanzanica, and Navid Nobani. A survey on xai and natural language explanations. *Information Processing & Management*, 60(1):103111, 2023.
3. Stuart Russell. *Human compatible: AI and the problem of control*. Penguin Uk, 2019.

4. Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.
5. Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*, 2019.
6. Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. Perturbed masking: Parameter-free probing for analyzing and interpreting bert. *arXiv preprint arXiv:2004.14786*, 2020.
7. Sandipan Sikdar, Parantapa Bhattacharya, and Kieran Heese. Integrated directional gradients: Feature interaction attribution for neural nlp 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 865–878, 2021.
8. Soumya Sanyal and Xiang Ren. Discretized integrated gradients for explaining language models. *arXiv preprint arXiv:2108.13654*, 2021.
9. Joseph Enguehard. Sequential integrated gradients: a simple but effective method for explaining language models. *arXiv preprint arXiv:2305.15853*, 2023.
10. Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-attention attribution: Interpreting information interactions inside transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12963–12971, 2021.
11. Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. Towards faithful model explanation in nlp: A survey. *Computational Linguistics*, pages 1–67, 2024.
12. Enja Kokalj, Blaž Škrlj, Nada Lavrač, Senja Pollak, and Marko Robnik-Šikonja. Bert meets shapley: Extending shap explanations to transformer-based classifiers. In *Proceedings of the EACL hackashop on news media content analysis and automated report generation*, pages 16–21, 2021.
13. Jesse Vig. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*, 2019.
14. Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. exbert: A visual analysis tool to explore learned representations in transformers models. *arXiv preprint arXiv:1910.05276*, 2019.
15. Oren Barkan, Edan Hauon, Avi Caciularu, Ori Katz, Itzik Malkiel, Omri Armstrong, and Noam Koenigstein. Grad-sam: Explaining transformers via gradient self-attention maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2882–2887, 2021.
16. Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. *arXiv preprint arXiv:2101.00288*, 2021.
17. Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025, 2020.
18. Siddhant Garg and Goutham Ramakrishnan. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*, 2020.
19. Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

20. Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. *Advances in neural information processing systems*, 31, 2018.
21. Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*, 2020.
22. Shiyuan Huang, Siddarth Mamidanna, Shreedhar Jangam, Yilun Zhou, and Leilani H Gilpin. Can large language models explain themselves? a study of llm-generated self-explanations. *arXiv preprint arXiv:2310.11207*, 2023.
23. Yanda Chen, Chandan Singh, Xiaodong Liu, Simiao Zuo, Bin Yu, He He, and Jianfeng Gao. Towards consistent natural-language explanations via explanation-consistency finetuning. *arXiv preprint arXiv:2401.13986*, 2024.
24. Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31, 2018.
25. Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*, 2019.
26. Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. In *The 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL 2023)*, 2023.
27. Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.
28. Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.
29. Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.
30. Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9, 2024.
31. Peter West, Chandra Bhagavatula, Jack Hessel, Jena D Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. Symbolic knowledge distillation: from general language models to commonsense models. *arXiv preprint arXiv:2110.07178*, 2021.
32. Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.
33. Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*, 2023.
34. Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
35. Consumer Financial Protection Bureau. HMDA National Loan Level Dataset 2022, 2022.

36. Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.
37. Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
38. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.