Understanding Hallucinations in LLMs: A Graph-Based Reproduction Study

1 Introduction

Large Language Models (LLMs) have recently surged in popularity and have a wide range of applications in different tasks. However, even though the outcomes of LLMs are fast, relatively reliable, and human-like, they may generate hallucinations in outcomes, which raises concern about the trustworthiness of the information these models provide. The research [1] proposes a graph-based architecture for hallucination detection. Researchers create a graph structure connecting generations that lie closely in the embedding space. Moreover, they employ a Graph Attention Network which utilizes message passing to aggregate information from neighbouring nodes and assigns varying degrees of importance to each neighbour based on their relevance. In our project, we test the performance of different thresholds of similarity of embeddings, and evaluate the generalizability of the hallucination detection proposed in the research paper by applying it to a new dataset and LLM.

2 Literature review (paper)

2.1 Motivation

The motivation for this study is based on two key premises. First, LLM hallucinations are not random but share structural characteristics in the latent space. Second, based on the principle of homophily, which is that similar entities cluster together, researchers investigate whether hallucinated texts cluster in embedding space due to shared characteristics. Based on the outlined premises and assumptions, researchers propose leveraging graph structures and message passing to reveal underlying patterns in the data.

2.2 Objectives

From the 2 hypotheses, researchers formulate the following research questions to answer.

1. Do LLM-generated hallucinations share characteristics?

2. Can we leverage graph structures to identify and learn these characteristics?

3. If learned, can we use this knowledge to identify hallucinations among new incoming LLM generations through label recovery?

3 Methodology

The methodology aims to detect hallucinations generated by large language models using graph structures. It consists of five key steps: data generation, graph construction, training a graph attention network, performing the label recovery task, and evaluation. These steps are illustrated in the following *Figure 1*.

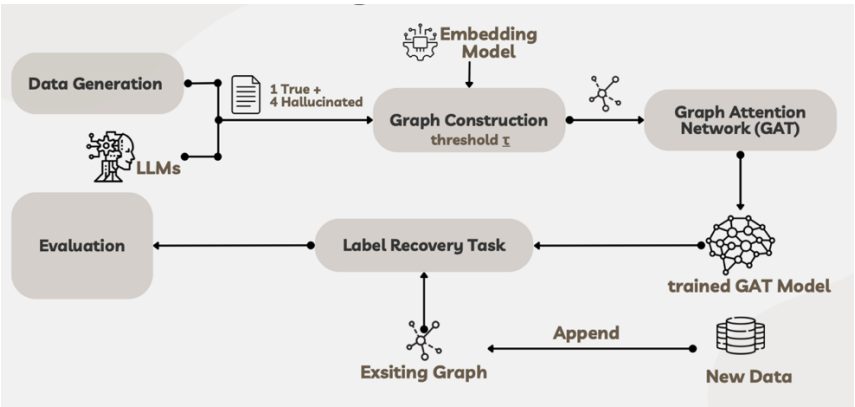

Figure 1 The diagram of methodology

3.1 Data Generation

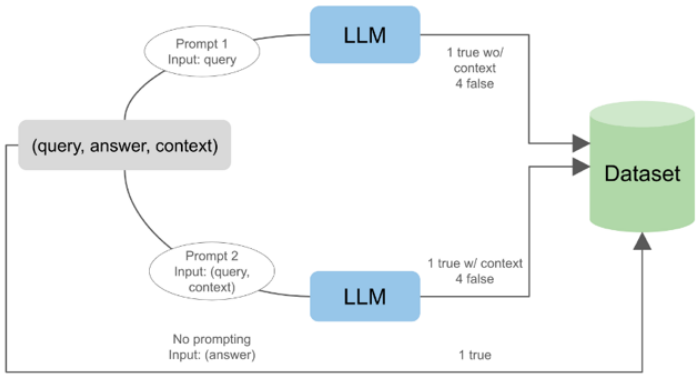

Figure 2 The Diagram of Data Generation Process

Detecting hallucinations requires a dataset with clear labels distinguishing grounded content from hallucinated content, which existing datasets fail to capture effectively. To address this, a new dataset was created. Hallucinations were defined as misleading sentences, prompting the model to generate challenging in-context hallucinations. Starting with a question-answering dataset, an LLM was prompted to generate both true and misleading responses. For each query, the model produced 11 statements: one correct answer from the dataset, two true statements generated by the model (one with context and one without), and eight intentionally misleading ones. This approach ensured a balanced dataset with clearly labelled grounded and hallucinated content. The process is shown on the *Figure 2* above.

3.2 Graph Construction

After generating the dataset, a graph structure is constructed to enable further training. Each statement is represented as a node, with an embedding model mapping these nodes into a latent space. Nodes are connected based on cosine similarity between their embeddings: if the similarity exceeds a threshold (τ), an edge is created. This method reduces computational complexity while maintaining meaningful relationships, clustering similar statements—hallucinated with hallucinated, and true with true.

The choice of τ is critical to balancing graph connectivity. Ideally, the node degree distribution should be relatively uniform, avoiding an overabundance of either highly connected or isolated nodes.

3.3 Graph Attention Network (GAT) Training

The Graph Attention Network (GAT) is the core of our framework, designed to detect hallucinations by leveraging graph structures. Nodes represent sentence embeddings with reduced dimensions using an MLP, and edges connect semantically similar nodes based on cosine similarity.

The GAT employs an attention mechanism to dynamically weigh neighbors, enabling the model to focus on the most relevant ones. For each node, it aggregates information from its neighbors via attention-weighted sums, uncovering patterns that differentiate hallucinated from true statements. This process enhances the embeddings by incorporating local and global context. This approach effectively captures latent patterns to distinguish hallucinations.

3.4 Label Recovery Task

After training, the GAT is tested for its ability to generalize to unseen data. New statements are added to the graph as nodes and connected to existing nodes based on semantic similarity (using the same cosine similarity threshold, τ). The GAT performs label recovery, predicting whether these new statements are hallucinated by leveraging information from their connected neighbours.

This process highlights the robustness of the graph-based approach, as the GAT dynamically integrates new data into the existing graph structure without retraining. By aggregating information from similar nodes, the model generalizes effectively, identifying hallucinations in unseen statements while maintaining computational efficiency. This adaptability demonstrates the GAT's strength in handling evolving datasets.

4 Experiments & Conclusion

To address the research questions posed, experiments were conducted to assess the efficacy of their framework in detecting hallucinations. Additionally, they modified the number of labels to three and two, evaluating performance on two benchmarks, FEVER and SelfCheck-GPT, to gauge its applicability to other datasets.

4.1 Experiment Setting

2000 data points were sampled from *MSMARCO-QA* and employed to create training data using Meta's instruction-tuned Llama2. The generated data were partitioned at the sentence level into training (70%), validation (15%), and testing (15%) sets. For graph training, BERT served as the embedding model, along with an empirically selected threshold (0.85) and Binary Cross Entropy (BCE) as the loss function.

To evaluate the framework comprehensively, three metrics were utilized: (1) Macro-recall for assessing accuracy in identifying individual classes; (2) Macro-precision for measuring

prediction accuracy per class; and (3) AUC-PR for evaluating binary classification performance.

4.2 Ablation Study

To evaluate their framework, two baselines were initially chosen for comparison with GAT. One baseline utilized DeBERTa to embed query-answer pairs in a uniform encoder, while the other involved a three-layer MLP with ReLU and DeBERTa as the embedding model to encode the same data. The effectiveness of GAT was validated by its superior performance (Figure 3 left).

Split	Model	Recall	Precision	AUC-PR
Train	GAT	**0.5069**	**0.5844**	**0.4153**
	DeBERTa-QA	0.3882	0.5404	0.3517
	MLP-QA	0.3214	0.3880	0.2718
Val	GAT	**0.4972**	**0.5717**	**0.4096**
	DeBERTa-QA	0.3206	0.5059	0.3357
	MLP-QA	0.3150	0.3622	0.2953

Split	Model	Recall	Precision	AUC-PR
Train	GAT	0.5069	0.5844	0.4153
	CL + GAT	**0.8244**	**0.8281**	**0.7118**
	MLP-A	0.2512	0.3123	0.2014
	CL + MLP-A	0.4286	0.5892	0.3987
Val	GAT	0.4972	**0.5717**	0.4096
	CL + GAT	**0.5305**	0.5438	**0.4212**
	MLP-A	0.2256	0.3110	0.2057
	CL + MLP-A	0.3589	0.4956	0.3278
	kNN	0.2434	0.1895	0.2494

Figure 3 Result of ablation study

However, this initial experiment revealed the limited discriminative power of BERT embeddings. This limitation arises because BERT primarily emphasizes contextual, syntactic, and semantic aspects rather than "validity" or "truthfulness" [1]. To introduce more comprehensive comparisons, a contrastive learning layer was incorporated. Subsequent experiments were conducted to compare the framework with and without CL, alongside an additional baseline utilizing a two-layer MLP with ReLU to encode answers exclusively. The results (Figure 3 right) further supported this hypothesis.

These experiments also addressed their primary research question: "Do LLM-generated hallucinations exhibit shared characteristics?" Their framework successfully discerned hallucinations by identifying underlying features, thus providing an affirmative answer to this question.

To address the second research question: "Can we leverage graph structures to identify and learn these characteristics?" the authors compared GAT with kNN (k-Nearest Neighbour). The outcome (Figure 3 right) indicated that graph structures excel in modeling such characteristics, thereby affirming the utility of this approach.

4.3 Performance on Test Set

Addressing the third research question: "If learned, can we use this knowledge to identify hallucinations among new incoming LLM generations through label recovery?" the framework was assessed on the test set. Similar performance (refer to Figure 4) demonstrated the method's efficacy in identifying hallucinations within new data entries.

	Recall	Precision	AUC-PR
CL + GAT	**0.5142**	0.5430	**0.4057**
GAT	0.4830	**0.5603**	0.3887
CL + MLP-A	0.3727	0.5122	0.3419

Figure 4. Result on test set.

4.3.1 Generalizability

Method	Recall	Precision	Label Accuracy
CL + GAT	0.7079	**0.4712**	0.6471
UNC-NLP	**0.7091**	0.4227	**0.6821**

Figure 5 Result on FEVER.

Method	Sentence-level (AUC-PR)	
	NonFactual	Factual
Random	0.7296	0.2704
LLM + BERT Scores	**0.8196**	**0.4423**
CL + GAT	0.7799	0.4002

Figure 6. Result on SelfCheck-GPT.

To evaluate the framework's generalizability across different datasets, the researchers modified the number of labels to three and two and tested its performance on two benchmarks, FEVER and SelfCheck-GPT.

The proposed framework delivered comparable results to the leading architecture on the FEVER benchmark. However, on SelfCheck-GPT, although it seemed less competitive to the LLM + BERT Score, this discrepancy was attributed to the dataset's limited size. Notably, given that their method does not necessitate access to search-based approaches, the proposed framework exhibited sufficiently impressive performance.

4.4 Experiments (ours)

To delve deeper into the factors influencing the framework's performance, we conducted several experiments focusing on two key aspects: (1) graph training settings, which encompassed the similarity thresholds utilized for edge determination and the choice of embedding model. (2) training data settings, which included the dataset employed and the specific LLM utilized to generate hallucination data.

Throughout these experiments, we maintained consistency with the methodology outlined in the paper, employing the same pipeline (GAT+CL) and adhering to identical hyperparameters such as batch size, training epochs, and evaluation metrics (Macro-Recall, Macro-Precision, and AUC-PR).

4.4.1 Experiment 1: Threshold & Embedding Model

The original paper posited that the similarity threshold is dataset-dependent, to assess this assertion, 60% (i.e. 1200 data samples) of the original training data was used.

Considering that the similarity is computed using the cosine similarity on the embeddings generated by the English uncased version of BERT [2] another embedding model, DeBERTa [3], was used to compare the threshold and performance. This version was chosen because it was also used in their two baselines, DeBERTa-QA and MLP-QA. For each embedding model, 8 values of threshold were selected around the original optimal value (0.85) in terms of all the training data.

Table 1 reveals that both the dataset size and the choice of embedding model influence the optimal threshold values and overall performance. Analysis of the BERT statistics indicates that with a smaller dataset, the optimal threshold tends to decrease. Optimal values in the range of 0.60 to 0.65 outperform the original 0.85 threshold in the new setting.

Interestingly, DeBERTa exhibits inferior performance compared to BERT, showing instability as depicted in the line charts (Figure 7) and corroborated by Table 1, where the optimal thresholds for the three metrics vary significantly.

This outcome seems contradictory to the expectation that DeBERTa, being a larger pre-trained model than BERT, would yield superior results. However, the original paper has expounded on why DeBERTa was excluded in the MLP-A baseline when evaluating contractive learning, while being utilized in the DeBERTa-QA and MLP-QA baselines. The rationale was anchored in the potential distribution shift when embedding answers only, given DeBERTa's distinct embeddings. This distributional incongruence poses challenges during graph training, aligning with the subpar performance observed in this experiment.

Embedding Model	Threshold	Recall	Precision	AUC-PR
BERT (CL+ GAT) (60% data)	0.60	**0.5246**	0.5884	**0.4334**
	0.65	0.5074	**0.6110**	<u>0.4305</u>
	0.70	<u>0.5108</u>	0.5728	0.4246
	0.75	0.4266	<u>0.5893</u>	0.3863
	0.80	0.5100	0.5398	0.4118
	0.85	0.4660	0.5858	0.4025
	0.90	0.4727	0.5508	0.3968
	0.95	0.4675	0.5302	0.3894
DeBERTa (CL+ GAT) (60% data)	0.60	0.4675	0.5302	**0.3894**
	0.65	0.4423	0.5606	0.3555
	0.70	0.4682	0.5622	0.3679
	0.75	0.4487	**0.5973**	0.3633
	0.80	0.3241	0.4532	0.2888
	0.85	<u>0.4842</u>	0.5169	<u>0.3736</u>
	0.90	0.4459	<u>0.5699</u>	0.3710
	0.95	**0.4952**	0.5221	0.3732

Table 1: Result of evaluation. Highest values for each threshold and model are bold, and the second best are underlined.

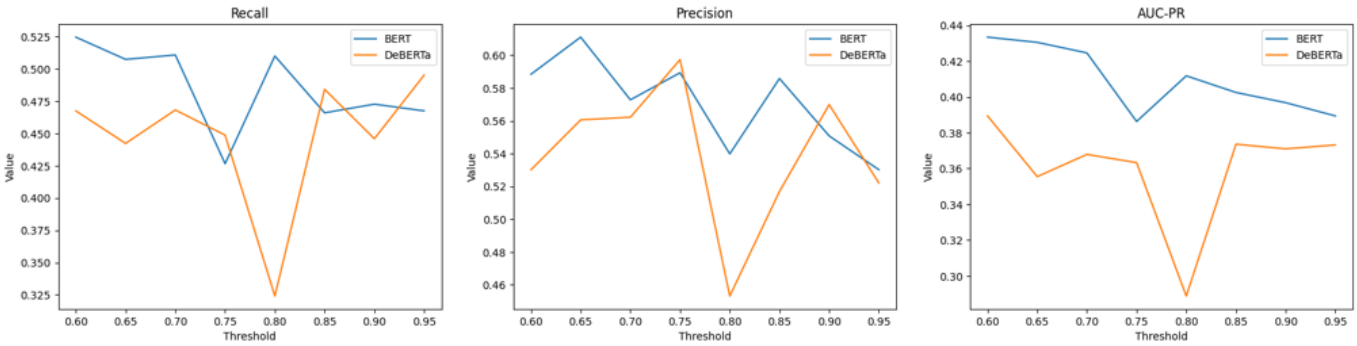

Figure 7. Line charts of the experiments regarding threshold and embedding model.

4.4.2 Experiment 2: Large Language Model

Large Language Models (LLMs) play a critical role in the study of hallucination detection. To further evaluate the robustness of our method, an experiment using Qwen-2.5-14B [4] was conducted. We compared its results with Llama2-13B, which is the original model for data generation, and the results is as shown on the *Table 2*.

LLMs	Recall	Precision	AUC-PR
Qwen2.5-14B	**0.5434**	**0.6240**	**0.4210**
Llama2-13B	0.4660	0.5858	0.4025

Table 2 The LLM Experiment Results

The results showed that Qwen2.5 outperforms Llama2 in all metrics. This suggests that the quality of data generation plays a significant role in downstream hallucination detection. Qwen's ability to generate more semantically distinct and contextually grounded statements may have contributed to these improvements.

4.4.3 Experiment 3: Dataset

To evaluate the capability of generalization of this method on another dataset, we changed the dataset used. The new dataset, Scientific Question Answering dataset (Sci_Q) [5] contains 13,679 crowdsourced science exam questions about Physics, Chemistry and Biology. An additional paragraph with supporting evidence for the correct answer is provided. We use the supporting evidence as the context. Other settings and the are the same as default. number of samples are same as our previous experiments. The results are shown in Table 3.

Dataset	Recall	Precision	AUC-PR
Sci_Q	**0.3353**	**0.4454**	**0.3030**
MSMARCO-QA	0.4660	0.5858	0.4025

Table 3 The Dataset Experiment Results

We can observe that the performance on the Sci_Q dataset does not reach the levels observed with the MSMARCO-QA dataset. We analyse that the brevity of answers in the Sci_Q dataset may contribute to the disparity in performance. the answers in Sci_Q are

shorter, resulting in hallucinated statements that deviate minimally from the true statements, often by only a few words. This minimal deviation may challenge the model's capacity to effectively distinguish between true and hallucinated statements.

We also visualize the embedding before and after the contrastive learning. The result is shown in Figure 8. we can recognize the cluster of false statements, represented by red dots, after the contrastive learning. The figure shows that contrastive learning helps in distinguishing embeddings and leads to better performance.

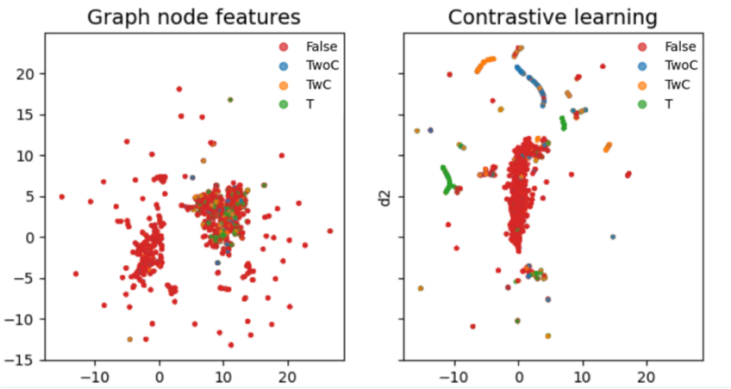

Figure 8 Projection of the dataset before and after contrastive learning

5 Conclusion

From our experiments, we explore the better range of threshold value for two models, BERT and DeBERTa. We change the LLM and dataset to evaluate the generalization capability. We find that the method outperforms on LLM that can generate more semantically distinct and contextually grounded statements. However, upon modifying the dataset, it was observed that the model's performance on the Sci_Q with shorter questions and answers did not match the efficacy demonstrated with the MSMARCO-QA.

The research reveals LLM-generated hallucinations share characteristics and shows the potential of GAT in LLM hallucination detection and the power of contrastive learning. The model will have more satisfactory performance with thresholds ranging from 0.60~0.65, with more powerful LLM that can generate more semantically distinct and contextually grounded statements and datasets with more open questions with longer answers.

6 References

[1] N. Nonkes, S. Agaronian, E. Kanoulas, and R. Petcu, “Leveraging Graph Structures to Detect Hallucinations in Large Language Models,” *arXiv.org*, Jul. 05, 2024. https://arxiv.org/abs/2407.04485

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv.org, Oct. 11, 2018. https://arxiv.org/abs/1810.04805

[3] P. He, X. Liu, J. Gao, and W. Chen, “DeBERTa: Decoding-enhanced BERT with Disentangled Attention,” arXiv.org, Jun. 05, 2020. https://arxiv.org/abs/2006.03654
[4] J. Bai et al., “Qwen Technical Report,” arXiv.org, Sep. 28, 2023. https://arxiv.org/abs/2309.16609
[5] Johannes Welbl, Nelson F. Liu, Matt Gardner, SciQ: "Crowdsourcing Multiple Choice Science Questions", proceedings of the Workshop on Noisy User-generated Text (W-NUT) 2017. https://arxiv.org/abs/1707.06209