Leveraging Graph Structures to Detect Hallucinations in Large Language Models

Noa Nonkes*, **Sergei Agaronian***, **Evangelos Kanoulas**, **Roxana Petcu**

University of Amsterdam

noanonkes@gmail.com, r.m.petcu@uva.nl

Abstract

Large language models are extensively applied across a wide range of tasks, such as customer support, content creation, educational tutoring, and providing financial guidance. However, a well-known drawback is their predisposition to generate hallucinations. This damages the trustworthiness of the information these models provide, impacting decision-making and user confidence. We propose a method to detect hallucinations by looking at the structure of the latent space and finding associations within hallucinated and non-hallucinated generations. We create a graph structure that connects generations that lie closely in the embedding space. Moreover, we employ a Graph Attention Network which utilizes message passing to aggregate information from neighboring nodes and assigns varying degrees of importance to each neighbor based on their relevance. Our findings show that 1) there exists a structure in the latent space that differentiates between hallucinated and non-hallucinated generations, 2) Graph Attention Networks can learn this structure and generalize it to unseen generations, and 3) the robustness of our method is enhanced when incorporating contrastive learning. When evaluated against evidence-based benchmarks, our model performs similarly without access to search-based methods.[1]

1 Introduction

Large Language Models (LLMs) have recently surged in popularity, notably due to the emergence of agents and models such as ChatGPT, Bard, Vicuna, and LLaMA (Brown et al., 2020a; Pichai, 2023; Ji, 2023; Touvron et al., 2023). Despite their increased capabilities for complex reasoning (Brown et al., 2020a), substantial challenges persist in grounding LLM generations to verified real-world knowledge. Ensuring that LLM generations are not only plausible but also factually correct poses a complex problem (Xu et al., 2024), which can be minimized (Liang et al., 2024) but so far not eliminated. Needless to say, even though LLMs possess an unparalleled ability to produce fast, credible, and human-like output, they are prone to hallucination (Ji et al., 2023a; Kasneci et al., 2023). This challenge expresses the non-trivial need for robust methods that detect and mitigate the spread of LLM-generated hallucinations.

Motivation The first underlying premise of this study is that LLM hallucinations are not unstructured, i.e., hallucinations share characteristics in the latent space. While extensive work has been done to mitigate LLM hallucinations (Ji et al., 2023b; Feldman et al., 2023; Martino et al., 2023), identifying these by their structural properties remains largely unexplored. Manakul et al. (2023) make a first step into identifying model-agnostic hallucinations through their SelfCheckGPT method, reliant solely on black-box access to the model to infer new generations. This method aligns with the idea that, given the same query, non-hallucinated samples exhibit a higher degree of similarity with each other than with hallucinated samples. We aim to extend this exploration by analyzing if, **independent** of the query, hallucinations share a higher degree of similarity with each other than with non-hallucinated generations. While SelfCheckGPT employs an implicit approach to model consistencies, we aim to do it explicitly by exploring semantic correspondences between hallucinations in the latent space. The second premise relies on the principle of homophily, which expresses that entities that share similar characteristics are more likely to form connections with each other (Zhang et al., 2016). In the context of hallucinations and text representations, homophily suggests that samples that share text-level characteristics tend to lie closer in the embedding space. We study if the degree of hallucination is such a characteristic.

*Equal contribution.

[1]The full code can be found on our GitHub repository.

Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing, pages 93–104

Based on the outlined premises and assumptions, we propose leveraging graph structures and message passing to reveal underlying patterns in the data. Comparing sentences in the embedding space involves assessing pairwise similarities between any pair of sentences. This results in $N(N-1)/2$ computations for N sentences, which is not computationally expensive for a one-time calculation, however, applying a neural network on top of this structure does not scale computationally, even with a simple single-layer feed-forward network. However, by leveraging the principle of homophily, we can form a graph where only similar nodes will have a direct connection between them, significantly reducing computational costs.

Objectives This study proposes two hypotheses: 1) LLM hallucinations arise from a pattern, which reflects in shared characteristics in the embedding space, and 2) we can efficiently leverage these characteristics using graph structures. We can formulate the following research questions:

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?

Contributions We introduce a hallucination detection framework for LLM-generated content. Given an existing dataset of hallucinations and true statements, we 1) leverage semantically rich sentence embeddings, 2) construct a graph structure where semantically similar sentences are connected, 3) train a Graph Attention Network (GAT) model that facilitates message passing, neighborhood attention attribution and selection, and 4) employ the GAT model to categorize new sentences as hallucinated or non-hallucinated statements.

According to our findings, 1) using semantic information to form connections between entities in the latent space helps to uncover links within hallucinated and non-hallucinated statements, 2) non-local aggregation enhances these links, 3) contrastive learning helps in distinguishing embeddings and leads to better performance, and 4) our method can accurately classify new unseen sentences as hallucinations or true statements.

The focus of this study is not getting on par performance with SOTA. Instead, we bring new hypotheses on the characteristics of LLM hallucinations. We implement and experiment with our method on multiple datasets: 1) we generate our own dataset by prompting an LLM to generate both true and misleading statements given a query and a context, and 2) we apply our framework to existing benchmark datasets to evaluate its performance on non-controlled data. Comparisons with benchmarks such as (Manakul et al., 2023; Thorne et al., 2018a) show that our method achieves close performance. Notably, we do not need access to external knowledge, LLM logits, or additional inference passes to the LLM, while keeping computational costs minimal.

We hypothesize that the latent space holds rich information beyond features such as contextualized, syntactic, and semantic information, for which the embeddings have been previously trained to capture. We also hypothesize that this information can be discovered and leveraged using geometric information. We propose a method that can be extended beyond the hallucination problem, and which can be generalized, applied to, and experimented with using any categorical label.

2 Related Work

LLMs The field of Natural Language Processing (NLP) has seen a significant evolution, from early probabilistic approaches such as Naive Bayes (Kim et al., 2006) to transformer models (Wolf et al., 2020) with attention mechanisms (Vaswani et al., 2017). This evolution also leads to LLMs which play a significant role in NLP tasks and applications (Alec et al., 2019; Brown et al., 2020b), yet they face a significant challenge known as *hallucination generation* that has been thoroughly studied.

Prompt verification SelfCheckGPT (Manakul et al., 2023) mitigates hallucinations using a sampling-based approach that facilitates fact-checking in a zero-resource fashion. The authors leverage the idea that if an LLM has knowledge on a certain subject, true generations from the same query are likely to be similar and factually consistent. They compare multiple query-dependent generated responses to identify fact inconsistencies indicative of hallucinations. Instead of only retrieving the most likely generated sequence of the model, they draw N further stochastic LLM responses and query the model itself to ascertain whether each sample supports the hallucination.

In essence, SelfCheckGPT does not need external knowledge but utilizes its internal knowledge to self-detect structural aspects of hallucinations. This approach, called prompt verification, achieves a notable 67% AUC-PR score in factual knowledge classification, showing potential for zero-shot fact-checking. However, it involves computational overhead as sampling LLM generations requires multiple forward passes to classify each single statement. Outside of SelfCheckGPT, multiple other approaches leverage prompt verification (Dhuliawala et al., 2023; Varshney et al., 2023).

Retrieval-based The Fact Extraction and Verification (FEVER) Shared Task (Thorne et al., 2018a) brings together several other approaches to hallucination detection. The task participants were challenged to classify whether human-written facts can be supported or refuted while having access to documents retrieved from Wikipedia. The task is mostly split into three parts. For document selection, many teams adopt a multi-step approach, which typically involves techniques such as Named Entity Recognition (Shaalan, 2014), Noun Phrases (Zhang et al., 2007), and Capitalized Expression Identification. The results are then used as inputs for querying a search API such as Wikipedia. The next step involves extracting relevant sentences through methods such as keyword matching, supervised classification, and sentence similarity scoring. Finally, for natural language inference, the extracted evidence sentences are often concatenated with the claim and passed through models such as a simple multilayer perceptron (MLP), Enhanced LSTMs (Chen et al., 2017), or encoder models to synthesize and evaluate the relationships between them. One notable difference between FEVER models and previously described work (Manakul et al., 2023) is that they have access to external sources.

Benchmark datasets TruthfulQA (Lin et al., 2022) proposes a benchmark for analyzing how accurate a language model is in generating answers given a question. The benchmark includes 817 questions spanning over 38 categories, which require a wide range of reasoning capabilities, such as questions in health, law, finance, and politics. Moreover, the questions are crafted in a manner that could lead humans to provide incorrect answers due to false beliefs or misconceptions. In this work, the authors analyze the performance of models such as GPT-3, GPT-Neo, GPT-J, GPT-2 and T-5 (Brown et al., 2020b; Black et al., 2022; Alec et al., 2019; Raffel et al., 2020), identifying that the best model was truthful on only 56% of the generations, while human performance reaches 94%. Compared with the other hallucination benchmarks, the correctness of an answer in TruthfulQA can only be assessed in association with its query. Therefore, we do not consider these answers as hallucinations on their own. While we could model this dataset by merging all answers with their associated queries, that would induce a major bias in the semantic similarity calculations when forming our method's graph structure. As a result, we do not evaluate our method on TruthfulQA but focus on datasets where hallucinations can be detected on the answer level only.

Other hallucination detection methods There are numerous alternative approaches. Luo et al. (2023) tests the familiarity of the LLM with the query prior to generation. The model withholds generation if the familiarity is low. Other studies look into Bayesian estimation in retrieval-augmented generation. Wang et al. (2023) achieves an AUC-PR of around 62% for factual knowledge, but introduces additional time and compute due to reliance on a search engine for external evidence retrieval. Another approach (Chen et al., 2023) aims to detect hallucinations through training a discriminator on the RelQA LLM-generated question-answering dialogue dataset. Their method achieves 85.5% accuracy on automatic labels and 82.6% AUC-PR on human labels, although reliance on human annotations introduces ambiguity.

In comparison to previous work, our method does not require access to external knowledge, nor to the LLM used for generating data, avoids biases associated with additional prompting, and eliminates costs associated with further inference.

3 Methodology

Assume a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$ consisting of n samples, where x_i denotes a sentence and y_i represents an ordinal categorical label indicating the degree of hallucination of x_i.

3.1 Graph Construction

Consider a model $\phi(x) = e$, where $e \in \mathbb{R}^{768}$, which maps x_i to its sentence-level embedding representation e_i. We construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as follows:

- $\mathcal{V}$ is the set of nodes. Each node $v \in \mathcal{V}$ corresponds to a single data point x_i from dataset D. More precisely, the features of a node consist of the sentence-level embedding $\phi(x_i)$. We employ BERT (Devlin et al., 2019) as the model ϕ to transform textual representations into embeddings.

- Two utterances $\{u, v\} \in \mathcal{V}$ are connected with an edge $(u, v) \in \mathcal{E}$ if and only if the semantic similarity between nodes u and v exceeds a threshold τ. The semantic similarity between two utterances is calculated using cosine similarity.

The choice of τ must ensure a balance in graph connectivity. Ideally, the node degree distribution should be relatively uniform, with a limited number of both highly connected and disconnected nodes. Additionally, we aim to avoid spikes in node degrees, as they may indicate the formation of hubs. The value of τ is dependent on the D.

3.2 Graph Attention Network

We employ a GAT model (Veličković et al., 2018) on our semantically-driven graph structure $\mathcal{G}$. Computing attention scores over sentence embeddings involves significant computational costs due to their high-dimensional representation. Consequently, we first reduce the dimensionality of node features by training a basic MLP. Then, we apply GAT on the reduced node features. The model will learn to map the sentence embeddings to a label indicating its degree of hallucination. We choose to model this problem with graph structures for two primary reasons: 1) to aggregate information via message passing, driven by our intuition that sentences that exhibit similar degrees of factuality share common structural components, and 2) to leverage edge weights, such that the level of similarity influences the information shared through message passing is expected to influence the message passing mechanism. We formalize the problem as an ordinal regression task as follows:

- **Label Encoding**: If a data point x_i has associated label L, then it is classified into all lower labels. Let L be an ordinal label and encode(L) be the corresponding encoding. Then, we can define encode(L) as:

$$\text{encode}(L)_i = \begin{cases} 1 & \text{if } i \leq L \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where encode$(L)_i$ represents the ith element of the encoding vector.

By employing a graph-based model, we aim to validate our assumptions that hallucinations exhibit shared characteristics within the embedding space. We add connections between sentences that show a high degree of similarity and, during training, we exchange information between nodes and their local neighborhood.

3.3 Label Recovery Task

Assume a new evaluation dataset D'. We first append D' to the existing graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and then perform one forward pass to the trained GAT using the new graph $\mathcal{G}' = (\mathcal{V} \cup \mathcal{V}', \mathcal{E} \cup \mathcal{E}' \cup \mathcal{A})$ to solve a label recovery task. $\mathcal{A}$ represents the edges formed between the nodes V and V'. We define the label recovery task as follows:

- **Label recovery**: Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ the set of edges. The label recovery task involves inferring missing labels for a subset of nodes $\mathcal{U} \subseteq \mathcal{V}$ based on available information of the known labels of nodes $\mathcal{V} \setminus \mathcal{U}$.

3.4 Data Generation

We first apply our method to our own hallucination-generated dataset. Creating our own dataset allows for more control over the modeling choices and requirements of our methodology, such as degree of connectivity, homophily, and encoding techniques, ensuring a more targeted evaluation. However, we recognize the importance of generalizability, and therefore we also validate our approach on existing datasets to assess its performance across diverse contexts and benchmarks.

Prompt During the process of LLM-driven hallucination generation, we design a prompt that guides the model to create statements for a multiple-choice exam, as shown in Appendix A. In our prompt, we refer to *hallucinations* as *misleading statements*. This is a modeling choice we took because *hallucinations* and *misleading statements* are conceptually similar, however, by labeling them as *misleading* we guide the model to generate statements intended to deceive the reader into believing they are true. This approach ensures the generation of hard in-context hallucinations instead of general hallucinations.

Retrieval-based generation We construct our data using retrieval-augmented generation on a representative question-answer (QA) dataset. We first sample queries along with their corresponding answer and associated context. We then instruct the LLM to generate a multiple-choice exam, where queries act as exam questions. This technique orients the LLM to generate misleading statements alongside true statements for each prompting stage, as a form of conditional generation. The evaluation of the generated data is conducted solely through human assessment. Although the exam-instruction format facilitates the generation of misleading statements, we acknowledge that LLMs are susceptible to bias and hallucinations, which can be reflected in our generated dataset. Existing biases in the model might skew the types of hallucinations in an unintended way, however, we assume that the effects of LLM hallucinations when intentionally induced have minimal impact on our study. We use two prompting techniques for instructing the LLM to generate both true and misleading answers, given either 1) the query, or 2) the query with its associated context, both part of the QA dataset. The latter is aimed at obtaining context-aware true statements from the LLM.

We thus generate a dataset that contains, for each query, 11 statements: 1 true *extracted* from the QA dataset, 1 true without context *generated*, 1 true with context *generated*, and 8 hallucinated *generated* statements. The overview of this process is illustrated in Figure 1. We provide the prompts in Appendix A. Each statement is assigned a label $y_i \in [0, 1, 2, 3]$ representing hallucinated, true w/o context, true w/ context, and true statement. We solve a categorical regression task, and as such, the labels are ordinal one-hot encoded as presented in Section 3.2.

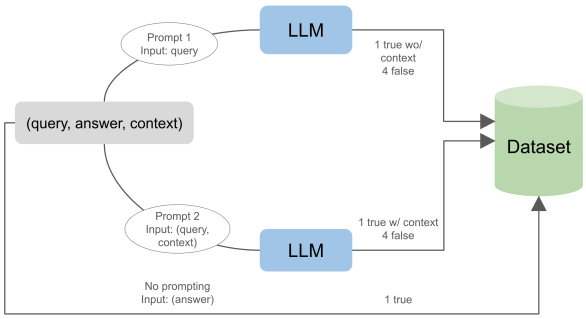

Figure 1: Data generation process.

4 Experimental Setup

4.1 Dataset

We used data from MSMARCO-QA (Nguyen et al., 2016), selecting 2000 questions within the biomedical domain (Xu et al., 2020) with answers consisting of a minimum of five words, to avoid sentences with explicit meaning only when paired with their corresponding question (examples are: "*Yes*", "*No*", "*Non-surgical*", or "*Virus infection*"). As mentioned in Section 3.4, each entry in the MSMARCO-QA dataset consists of the query, answer, and context. We use the queries and context to generate true w/ context, true w/o context, and hallucinated statements by prompting Meta's instruction-tuned Llama2 (Touvron et al., 2023) $13B$ model[2]. The dataset is randomly divided into three segments: training, validation, and test, with partitions of 70%, 15%, and 15%, respectively. The random division is done on the sentence level, therefore same query generations are not necessarily in the same data partition.

4.2 Graph

We use the English uncased version of BERT (Devlin et al., 2019)[3] for sentence embeddings. The graph is constructed over the entire dataset, with designated masks for the training, validation, and test sets. Edges are formed between nodes with cosine similarity above a threshold $\tau = 0.85$, which was selected empirically to strike a balance, ensuring a reasonable level of graph connectivity. The resulting undirected graph has approximately $26M$ out of a potential of $240M$ edges (fully-connected). Additionally, we use the cosine similarity values as edge attributes.

4.3 Graph Attention Network

The embeddings are reduced to a dimensionality of 32 features using a trained single-layer MLP. The GAT model incorporates a single graph attention convolutional layer, transitioning from 32 to 3 dimensions, with 2 attention heads. The GAT model explores and aggregates information from the immediate neighborhood based on the attention mechanism.

[2]https://huggingface.co/OpenAssistant/llama2-13b-orca-8k-3319

[3]https://huggingface.co/bert-base-uncased

4.4 Baselines

We employ a three-layer MLP with ReLU activations, using the same BERT model for sentence embeddings, but with concatenated query-answer as inputs. As a second baseline, we employ a larger pre-trained language model DeBERTa (He et al., 2021)[4], processing query-answer pairs in a unified encoder for improved contextual understanding and predictions. The MLP offers a simple yet powerful method for analyzing relationships between the generations and respective queries and has been shown to be efficient in sentence-level analysis (Ramdhani et al., 2022; Akhtar et al., 2017). Meanwhile, DeBERTa is a complex transformer model that facilitates deeper comparisons of attention effects and structural differences between transformers and lightly connected graphs. Unlike our graph-based model, which relies solely on answer embeddings, the baselines utilize embeddings from both queries and answers, enhancing semantic expressiveness for better differentiation among classes.

4.5 Training

The model selection process is based on optimal macro-recall performance on the validation split, offering a comprehensive evaluation of the model's ability to identify instances across all classes, crucial in the context of highly imbalanced data. Training spans over 500 epochs utilizing the Adam optimizer (Kingma and Ba, 2017) with fixed learning rate 1×10^{-3}. The model is trained to solve a categorical regression task using Binary Cross Entropy (BCE) loss (Good, 1952). This choice reflects the need for nuanced penalization of misclassifications, where the model applies a lower penalty for misclassifying adjacent, compared to further-apart labels. Efforts are made to prevent data leakage between the data partitions. During training, weights corresponding to edges that connect nodes with either validation or test nodes are nullified, ensuring no information exchange (Equation 2). Backpropagation happens exclusively over the training nodes. Similarly, during validation, we modify edges connecting to test nodes (Equation 3). Recall is calculated exclusively on validation nodes. The full graph is used to assess performance on the test set (Equation 4).

$$\mathcal{N}(i) = \{j | (i,j) \in \mathcal{E}, i \in \mathcal{N}_{train} \text{ and } j \in \mathcal{N} \backslash \{\mathcal{N}_{val} \cup \mathcal{N}_{test}\}\} \quad (2)$$

$$\mathcal{N}(i) = \{j | (i,j) \in \mathcal{E}, i \in \mathcal{N}_{val} \text{ and } j \in \mathcal{N} \backslash \mathcal{N}_{test}\} \quad (3)$$

$$\mathcal{N}(i) = \{j | (i,j) \in \mathcal{E}, i \in \mathcal{N}_{test}\} \quad (4)$$

4.6 Evaluation Metrics

We calculate three metrics to evaluate the performance of our approach. Macro-recall assesses the model's accuracy in identifying individual classes, while macro-precision evaluates prediction accuracy per class. AUC-PR calculates the area under the precision-recall curve, providing a measure for binary classification performance. Additionally, these metrics are robust against class imbalance, making them suitable for evaluating our model on our imbalanced dataset.

4.7 Benchmark Datasets

In our study, we utilize two human-generated and annotated benchmarking datasets, namely FEVER (Thorne et al., 2018a) and SelfCheckGPT (Manakul et al., 2023). Our method can be generalized across other domains. To apply our method to another dataset, we re-construct the graph with the respective data partitions and redefine the labeling accordingly. These datasets are used for evaluating the performance of our models under conditions that mimic real-world scenarios. The specific methodologies employed for their use are discussed in Section 2.

The FEVER dataset (Thorne et al., 2018a) contains 185445 claims which are divided into three types of claims, namely *Supports*, *Refutes*, and *Not Enough Info*, each paired with evidence sentences. To apply our method to the FEVER dataset, we re-define the label as $y_i \in [0, 1, 2]$ and generate the graph based on the train/val/test partitions of FEVER. The participating models (Thorne et al., 2018a) formulate careful data processing approaches and make use of external sources to verify the factuality of the claims. If search-based evidence is found for a claim, it is classified as *Supports* or *Refutes*. Similarly, if no evidence is found, it is labeled as *Not Enough Info*.

The SelfCheckGPT dataset (Manakul et al., 2023) consists of 1908 sentences categorized as *Accurate*, *Minor inaccurate*, and *Major inaccurate*. To apply our method to SelfCheckGPT, we again redefine the labels as $y_i \in [0, 1, 2]$ and generate the graph by randomly splitting the data into train/val/test sets.

[4]https://huggingface.co/MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli

5 Results

5.1 Non-local Aggregation

To address our first research question, more specifically *1. Do LLM-generated hallucinations share characteristics?*, we analyze if our framework identifies an underlying structure of the embedding space. As shown in Table 1, GAT exhibits better performance compared to DeBERTa-QA and MLP-QA on all metrics. GAT has approximately 17% higher recall than both baseline models on the validation set. This suggests its superior ability to identify positive instances, reduce false positives, and effectively differentiate between true and hallucinated statements.

Table 1: Comparing performance between GAT, 3-layer MLP, and DeBERTa using query answer (QA). The best results for each metric and dataset split are highlighted in bold.

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

5.2 Contrastive Learning

Initial experiments revealed that BERT embeddings are not discriminative enough for our task. This is intuitively to be expected: we hypothesize that hallucinations share features in the latent space. However, this does not imply that these features are inherently discriminative within BERT embeddings, as BERT is trained to capture contextual, syntactic, and semantic information, rather than "validity" or "truthfulness".

To acquire enriched embeddings, we train a Contrastive Learning (CL) (Khosla et al., 2020) MLP on the train set. This choice aims to strengthen the model's ability to differentiate between classes. In CL, larger batch sizes often enhance performance by allowing more comparisons with negative examples, smoothing loss gradients. We found that a batch size of 256 suffices for good results. Extended training periods notably benefit CL. We train for 1000 epochs using a decoupled weight decay optimizer (Loshchilov and Hutter, 2019). Parameter group learning rates are set with a cosine annealing schedule (Loshchilov and Hutter, 2017). Our contrastive learned MLP (CL + MLP) consists of two linear layers: input size 768, sequentially transitioning to 768, and then to 128 with ReLU activation. After contrastive learning, the 32-dimensionality reduction MLP is applied.

5.3 Ablation Study

To assess the impact of incorporating CL, we compare the metrics of GAT with and without CL, alongside the MLP baseline. We train the MLP with CL to differentiate between answers only, leading to a new baseline MLP-A. This MLP is a two-layer model with hidden sizes 64 and 32, and ReLU activation. This comparison is excluded for the DeBERTa model, as MLP-A is trained solely on answers, and DeBERTa uses different embeddings, potentially leading to a distribution shift.

To further address our first research question, we evaluate the model's performance both with and without contrastive learning (CL). Table 2 reveals significant improvements in GAT's performance with CL, particularly a remarkable 32% increase in recall on the train set. While the validation set also shows overall improvements, there is a slight 3% dip in precision, countered by an approximate 3% increase in recall. Without CL, MLP's performance appears random. After using CL, there is an apparent improvement across all metrics. In particular, there is a 20% improvement in both precision and recall for the train set. Validation recall sees an approximate 10% increase, while precision increases by around 20%.

Table 2: Comparing performance between GAT, MLP, and kNN using contrastive learning (CL) versus without, with only answer (A) embeddings. For kNN we only show validation results. The best results for each metric and data split are highlighted in bold.

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

Despite CL significantly improving the results, the ablation study reveals it is not the only factor in improving performance. To answer our next

research question *2. Can we leverage graph structures to identify and learn these characteristics?*, we employ k-Nearest Neighbour (kNN) with CL-learned embeddings. We assess the independent expressiveness of these embeddings, anticipating that sufficiently robust embeddings would enable a reliable majority-voting mechanism. However, with $k = 5$, kNN shows consistent underperformance (detailed in Table 2). Further exploration involved training the same MLP as introduced in Section 5.2, which showed improved performance compared to MLP without CL-learned embeddings. However, the MLP still trailed the performance of GAT, with approximately a 20% decrease in validation recall, highlighting the significance of the graph structure. The attention mechanism of GAT is crucial in accurately identifying important neighbors. This refined approach which is solely reliant on spatial similarity outperformed the kNN method, highlighting the advantages of graph structures for efficient information propagation. Furthermore, edge masking ensures robustness by preventing information exchange between training and validation/test nodes during training. This method acts as a regularizer, enhancing the model's generalization capabilities (Rong et al., 2020).

5.4 Test Set Performance

Finally, to address the research question *3. If learned, can we use this knowledge to identify hallucinations among new incoming LLM generations through label recovery?*, we analyze the best-performing models by validation recall. The models that showcase the highest performance are GAT with and without contrastive learning. To ensure a fair comparison, we also consider the performance of the third-best model on the test set. The results are shown in Table 3. Performance on the test set reveals that GAT with CL outperforms the other models on every metric except precision. The GAT structure proves crucial for higher recall.

Table 3: Comparing performance on the test set between the best performing models: GAT, MLP with CL, and GAT without CL. The best results for each metric and data split are highlighted in bold.

	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

5.5 Generalizability on Other Benchmarks

We assess the generalizability of our method on two real-world datasets, namely FEVER (Thorne et al., 2018a) and SelfCheckGPT (Manakul et al., 2023). Section 2 discusses their original applications, while Section 4.7 details how we modify the labels for our model.

To benchmark our model's performance, we compare the results against the best performance of the first FEVER Shared Task challenge (Thorne et al., 2018b), shown in Table 4 as UNC-NLP. Our model outperforms UNC-NLP in precision, with accuracy being 4% lower. However, it is important to stress that, in comparison, we solve a closed-book problem, avoiding the computational overload and necessity of any external data or search-based model.

Table 4: For FEVER(Thorne et al., 2018a): Performance metrics on the FEVER dataset. The best results are highlighted in bold.

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

Following the methodology in SelfCheckGPT (Manakul et al., 2023), we use DeBERTa-large for sentence embeddings. Our model falls short of the pairwise consistency metrics computed using DeBERTa-large embeddings (with BERTScore), as demonstrated in Table 5. A plausible explanation is the dataset's small size. The method with BERTScore needs multiple LLM-generated statements, while our method, trained on a small set, requires more examples for effective learning.

Table 5: For SelfCheckGPT(Manakul et al., 2023): Factual sentences are labelled as *Accurate*, NonFactual sentences are labelled as *Major-* and *Minor-inaccurate*. AUC-PR scores for Random and w/ BERTScore are computed on the entire dataset; our method's scores are calculated on the test set. The best results are highlighted in bold.

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

6 Conclusion

This study shows the potential of GAT in LLM hallucination detection. Its adaptability and capabilities to find underlying graphical structures provide a significant advantage in discriminating between real and hallucinated generations. In the realm of hallucination detection, where information interconnects in complex ways, GATs' proficiency in navigating these connections proves invaluable.

Overall, this research reveals the pivotal role of structural information within graphs in discriminating between true and hallucinated statements. The incorporation of non-local aggregation serves to fortify these connections. The integration of a contrastive learned embedder enhances the discernment between true and hallucinated statements. Furthermore, this framework exhibits the capacity to extend beyond initial data, enabling generalization to real hallucinations.

Limitations and Future Work Several limitations to our approach should be considered: 1) it requires effort to model the data, create ordinal categorical labels, and construct the graph structure; 2) it does not allow for transparency at all; 3) the method is difficult to scale, as adding nodes involves an exponential number of embedding comparisons for adding edges in the graph. Moreover, GATs face a limitation when adding new nodes to the graph, hindering real-time classification. Addressing these limitations could be a focus for future work, with the exploration of dynamic graph attention networks (Shi et al., 2021) offering potential solutions. Dynamic GATs may facilitate the addition of new nodes, enabling real-time adaptation to evolving graph structures and addressing the current impracticality of real-time classification. Moreover, while the idea of this research was to model semantic relationships between individual utterances without explicitly assuming any connection between retrieval-based true statements and generations, it would be interesting to also simulate the query-answer baseline setting with the attention mechanism and analyze how the performance of our model changes.

References

Md Shad Akhtar, Abhishek Kumar, Deepanway Ghosal, Asif Ekbal, and Pushpak Bhattacharyya. 2017. A multilayer perceptron based ensemble technique for fine-grained financial sentiment analysis. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 540–546.

Radford Alec, Wu Jeffrey, Child Rewon, Luan David, Amodei Dario, and Sutskever Ilya. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog*, 1(8):9.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. Gpt-neox-20b: An open-source autoregressive language model. *CoRR*, abs/2204.06745.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Neelakantan Arvind, Shyam Pranav, Sastry Girish, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Tom B. Brown, Benjamin Mann, et al. 2020b. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 2020-Decem. Neural information processing systems foundation.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1657–1668. Association for Computational Linguistics.

Yuyan Chen, Qiang Fu, Yichen Yuan, Zhihao Wen, Ge Fan, Dayiheng Liu, Dongmei Zhang, Zhixu Li, and Yanghua Xiao. 2023. Hallucination detection: Robustly discerning reliable answers in large language models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 245–255, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1,

pages 4171–4186. Association for Computational Linguistics (ACL).

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *CoRR*, abs/2309.11495.

Philip Feldman, James R. Foulds, and Shimei Pan. 2023. Trapping LLM hallucinations using tagged context prompts. *CoRR*, abs/2306.06085.

Irving John Good. 1952. Rational decisions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 14(1):107–114.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DEBERTA: DECODING-ENHANCED BERT WITH DISENTANGLED ATTENTION. In *ICLR 2021 - 9th International Conference on Learning Representations*. International Conference on Learning Representations, ICLR.

Bin Ji. 2023. Vicunaner: Zero/few-shot named entity recognition using vicuna. *CoRR*, abs/2305.03253.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023a. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.*, 55(12):1–38.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023b. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 1827–1843. Association for Computational Linguistics.

Enkelejda Kasneci, Kathrin Sessler, et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*, volume 2020-December.

Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, and Sung-Hyon Myaeng. 2006. Some effective techniques for naive bayes text classification. *IEEE Trans. Knowl. Data Eng.*, 18(11):1457–1466.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Yuxin Liang, Zhuoyang Song, Hao Wang, and Jiaxing Zhang. 2024. Learning to trust your feelings: Leveraging self-awareness in llms for hallucination mitigation. *CoRR*, abs/2401.15449.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 3214–3252. Association for Computational Linguistics (ACL).

Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019*. International Conference on Learning Representations, ICLR.

Junyu Luo, Cao Xiao, and Fenglong Ma. 2023. Zero-resource hallucination prevention for large language models. *CoRR*, abs/2309.02654.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. *Preprint*, arXiv:2303.08896.

Ariana Martino, Michael Iannelli, and Coleen Truong. 2023. Knowledge injection to counter large language model (LLM) hallucination. In *The Semantic Web: ESWC 2023 Satellite Events - Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings*, volume 13998 of *Lecture Notes in Computer Science*, pages 182–185. Springer.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *CEUR Workshop Proceedings*, volume 1773. CEUR-WS.

Sundar Pichai. 2023. An important next step on our ai journey.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Yudi Ramdhani, Hanii Mustofa, Salman Topiq, Doni Purnama Alamsyah, Sandy Setiawan, and Leni Susanti. 2022. Sentiment analysis twitter based lexicon and multilayer perceptron algorithm. In *2022 10th International Conference on Cyber and IT Service Management (CITSM)*, pages 1–6. IEEE.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2020. Dropedge: Towards deep graph convolutional networks on node classification. *Preprint*, arXiv:1907.10903.

Khaled Shaalan. 2014. A survey of arabic named entity recognition and classification. *Comput. Linguistics*, 40(2):469–510.

Min Shi, Yu Huang, Xingquan Zhu, Yufei Tang, Yuan Zhuang, and Jianxun Liu. 2021. GAEN: Graph Attention Evolving Networks. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 1541–1547.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and verification (fever) shared task. *arXiv preprint arXiv:1811.10971*.

Hugo Touvron, Louis Martin, et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *Preprint*, arXiv:2307.09288.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *CoRR*, abs/2307.03987.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 5999–6009. Neural information processing systems foundation.

Petar Veličković, Arantxa Casanova, Pietro Liò, Guillem Cucurull, Adriana Romero, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR.

Xiaohua Wang, Yuliang Yan, Longtao Huang, Xiaoqing Zheng, and Xuanjing Huang. 2023. Hallucination detection for generative large language models by bayesian sequential estimation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 15361–15371. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Ying Xu, Xu Zhong, Antonio Jose Jimeno Yepes, and Jey Han Lau. 2020. Forget Me Not: Reducing Catastrophic Forgetting for Domain Adaptation in Reading Comprehension. In *Proceedings of the International Joint Conference on Neural Networks*. Institute of Electrical and Electronics Engineers Inc.

Ziwei Xu, Sanjay Jain, and Mohan S. Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *CoRR*, abs/2401.11817.

Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. 2016. Homophily, structure, and content augmented network representation learning. In *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, pages 609–618. IEEE Computer Society.

Wei Zhang, Shuang Liu, Clement T. Yu, Chaojing Sun, Fang Liu, and Weiyi Meng. 2007. Recognition and classification of noun phrases in queries for effective retrieval. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*, pages 711–720. ACM.

A Appendix

Figure 2: Prompt 1: Without provided Context

Task: {*Imagine you are crafting a multiple-choice exam in the field of biomedical studies.*}

{*Your task is to generate a set of statements related to a given question.*}

{*Provide one accurate statement as the correct answer (Answer 1) and four misleading statements that should appear as plausible distractors (Answers 2 to 5).*}

{*Ensure that the incorrect answers are not easily mistaken for accurate information related to the question.*}

Question: {*What is the role of the BRCA1 gene in breast cancer?*}

Context: {}

Figure 3: Prompt 2: With provided Context

. . . . . .

Task: {*Imagine you are crafting a multiple-choice exam in the field of biomedical studies.*}

{*Your task is to generate a set of statements related to a given question.*}

{*Provide one accurate statement as the correct answer (Answer 1) and four misleading statements that should appear as plausible distractors (Answers 2 to 5).*}

{*Ensure that the incorrect answers are not easily mistaken for accurate information related to the question.*}

Question: {*What is the role of the BRCA1 gene in breast cancer?*}

Context: {*The BRCA1 gene is a gene on chromosome 17 that produces a protein responsible for repairing DNA. Mutations in this gene can lead to reduced protein functionality, impairing DNA repair processes. This impairment increases the risk of mutations in other genes, which can result in uncontrolled cell growth and potentially lead to the development of breast cancer. The presence of mutated BRCA1 is a significant marker for an increased risk of breast and ovarian cancers in women, making genetic testing a key preventive measure for those with a family history of these cancers.*}

. . . . . .