GraphCheck: Breaking Long-Term Text Barriers with Extracted Knowledge Graph-Powered Fact-Checking

Anonymous ACL submission

Abstract

Large language models (LLMs) are widely used, but they often generate subtle factual errors, especially in long-form text. These errors are fatal in some specialized domains such as medicine. Existing fact-checking with grounding documents methods face two main challenges: (1) they struggle to understand complex multihop relations in long documents, often overlooking subtle factual errors; (2) most specialized methods rely on pairwise comparisons, requiring multiple model calls, leading to high resource and computational costs. To address these challenges, we propose ***GraphCheck***, a fact-checking framework that uses extracted knowledge graphs to enhance text representation. Graph Neural Networks further process these graphs as a soft prompt, enabling LLMs to incorporate structured knowledge more effectively. Enhanced with graph-based reasoning, GraphCheck captures multihop reasoning chains which are often overlooked by existing methods, enabling precise and efficient fact-checking in a single inference call. Experimental results on seven benchmarks spanning both general and medical domains demonstrate a 6.1% overall improvement over baseline models. Notably, GraphCheck outperforms existing specialized fact-checkers and achieves comparable performance with state-of-the-art LLMs, such as DeepSeek-V3 and OpenAI-o1, with significantly fewer parameters.[1]

1 Introduction

Large language models (LLMs) (Hurst et al., 2024; Dubey et al., 2024), have demonstrated powerful generative capabilities in various domains (Jiang et al., 2024; Lee et al., 2024; Liu et al., 2023a). However, due to limitations in training data and the lack of integration of domain-specific knowledge, LLMs often "*hallucinate*" factual errors or inaccurate information (McKenna et al., 2023; Zhang et al., 2023; Gao et al., 2023). As LLMs prioritize linguistic fluency and contextual relevance in their generation processes, the generated content may appear convincing while lacking factuality (Ramprasad et al., 2024; Huang et al., 2023). This issue is particularly concerning in specialized domains like medicine, where factual errors can compromise patient safety, leading to misdiagnoses, inappropriate treatments, and, in severe cases, life-threatening consequences (Yang et al., 2024b; Ahsan et al., 2023). Therefore, ensuring the reliability and factual accuracy of LLM outputs is essential (Yang et al., 2023; Liu et al., 2024c).

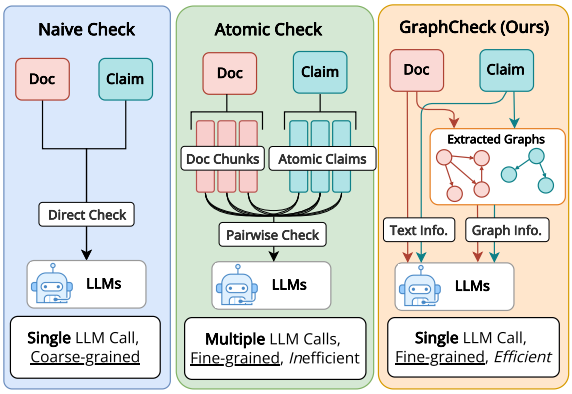

Figure 1: Comparison of fact-checking methods. **Naive Check** performs a single-pass evaluation but often misses detailed factual errors. **Atomic Check** ensures fine-grained verification by checking atomic facts individually but is inefficient due to multiple LLM calls. In contrast, our **GraphCheck** achieves fine-grained fact-checking in a single call, significantly improving efficiency while maintaining accuracy.

We consider the task of fact-checking claims against grounding documents, where the goal is to assess factual consistency based on provided textual evidence (Tang et al., 2024). Given the high cost and time demands of manual verification, modern fact-checking methods have shifted to automated approaches using LLMs or natural

[1] Our anonymous code is available at https://anonymous.4open.science/r/GraphCheck-1D43.

language inference (NLI) models (Fu et al., 2023; Kim et al., 2024). Standard LLM-based checking methods take a straightforward approach by directly feeding documents and claims into LLM for fact-checking judgment (Figure 1, left). However, when dealing with long-form documents, they often struggle to capture complex entity relations and overlook subtle inconsistencies given large volumes of information. Additionally, long prompts may exceed the LLM's context window, causing potential loss of relevant details and limiting the model from effective fact-checking. To address this, specialized methods (Zha et al., 2023; Min et al., 2023; Liu et al., 2024b) decompose long documents into smaller chunks and break claims into atomic facts, enabling fine-grained evaluation at the price of computational cost and efficiency (Figure 1, middle).

To address the problem of long text fact-checking, we propose ***GraphCheck*** (Figure 1, right), a graph-enhanced framework using extracted knowledge graphs (KGs) to capture multi-hop logical relations between entities, enhancing both global coherence and fine-grained understanding in long texts. We employ Graph Neural Networks (GNNs) to encode these graph structures and integrate the graph embeddings into LLM inputs. The direct comparison between the extracted document and claim graphs enables fine-grained factual verification in a LLM inference. The GNNs are trained on our curated general-domain synthetic graph data based on MiniCheck (Tang et al., 2024) training set, while the LLMs remain frozen. Empirically, we find that despite being trained on general-domain data, our model achieves improved performance not only on general-domain datasets but also on medical-domain datasets, demonstrating that its graph-enhanced reasoning ability generalizes across domains. We also provide this dataset as a benchmark for future research, allowing the training and evaluation of graph-based fact-checking.

In summary, our contributions are:

- **Pioneering Graph Reasoning for LLM Fact-Checking.** We propose GraphCheck, the first graph reasoning-enhanced LLM framework for fact-checking with grounding documents, ensuring fine-grained factual accuracy with high efficiency.
- **Enabling Fine-grained Explainability.** Our method enhances explainability by identifying the key entity relationships the model focuses on during fact-checking, ensuring a clear and verifiable reasoning process.
- **Providing a Benchmark for Graph-based Fact-Checking Models.** We introduce a synthetic dataset that pairs text with its corresponding extracted knowledge graph, enabling the training and evaluation of KG-enhanced fact-checking models.
- **Empirical Findings.** We demonstrate the effectiveness and efficiency of GraphCheck, achieving a 6.1% improvement over the base model in fact-checking across extensive general and medical benchmarks.

2 Related Work and Background

Methods in Detecting Hallucination. Recent fact-checking research (Yuan and Vlachos, 2023; Kim et al., 2023) use Retrieval-Augmented Generation (RAG) (Fan et al., 2024) and external knowledge bases like DBpedia (Lehmann et al., 2015) and Wikidata (Vrandečić and Krötzsch, 2014) to verify generated claims by retrieving structured or semi-structured data. Another line of research (Manakul et al., 2023; Mündler et al., 2023) focuses on verifying factual consistency using LLMs with grounding documents. These approaches harness LLMs' reasoning and language capabilities to fact-check claims against textual evidence. While effective for short texts, they often fail to capture fine-grained inconsistencies in longer documents, limiting their accuracy. Our work builds on this second setting, aiming to improve fact-checking performance on long texts by enhancing LLMs with structured graph-based reasoning.

Fact-Checking on Long Texts. To address the challenge of capturing detailed errors in long texts, recent methods have shifted towards using fine-grained units for fact-checking. Methods like FactScore (Min et al., 2023), MiniCheck (Tang et al., 2024), and ACUEval (Wan et al., 2024) focus on extracting atomic units from the generated text to enable fine-grained fact verification. However, these fine-grained fact-checking methods often require multiple calls to verify each unit or triple, especially for long texts, which greatly increases computational cost and time. In contrast, our approach uses KGs to model complex entity relationships in long texts, enabling fine-grained verification in a single call. This avoids repetitive calls and significantly improves efficiency.

Graph-based Methods for Enhancing Factuality.

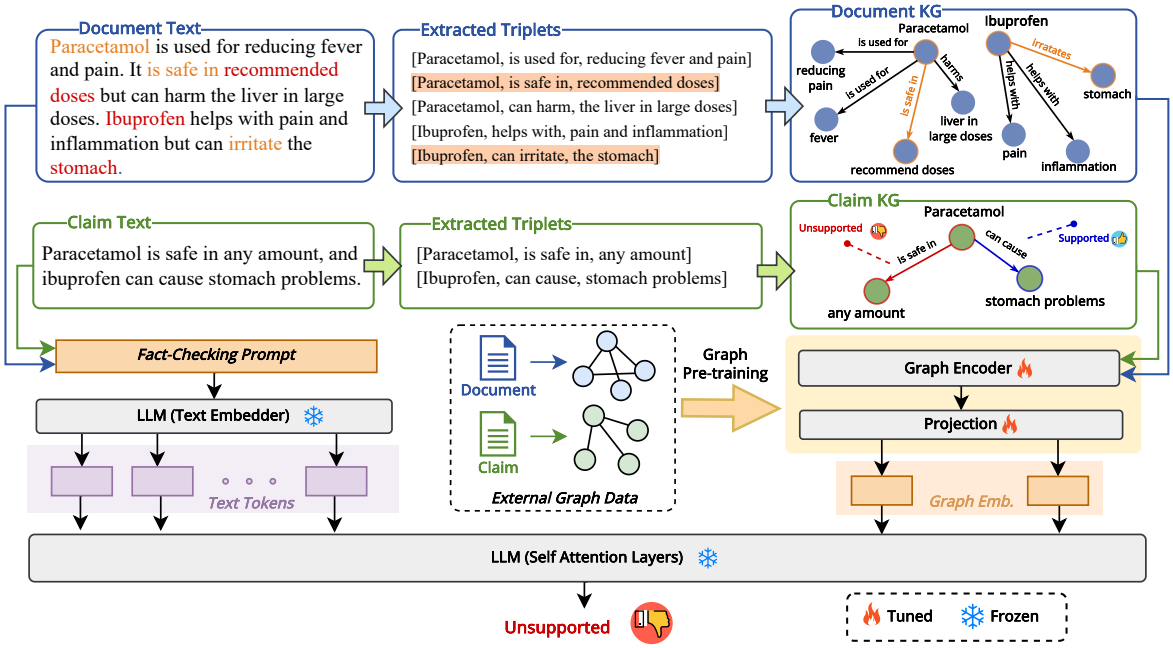

Figure 2: An illustration of the GraphCheck framework. Firstly, an LLM extracts entity-relation triples from both the claim and the document to construct KGs, respectively. A GNN pre-trained with external text graph data is then used to obtain graph embeddings from both KGs. These graph embeddings, combined with the text embeddings, are fed into an LLM for final fact-checking. This approach enables the LLM to perform fine-grained fact-checking by leveraging key triples in the KG (highlighted) alongside the text information.

Previous graph-based fact-checking methods have primarily focused on isolated triple evaluations or document-level encoding, often overlooking the global graph structure and topological information. GraphEval (Liu et al., 2024b) extracts triples from claims and evaluates their factual consistency individually using a pretrained NLI model. However, it also relies on pairwise comparisons and does not incorporate the overall graph structure, limiting its ability to capture complex relationships. FactGraph (Ribeiro et al., 2022) employs graph encoders to process documents and summary semantic graphs extracted via OpenIE. It then combines text and graph embeddings through an MLP for the final prediction. However, as a pre-LLM method, it lacks the powerful contextual reasoning ability of modern models. AMRFact (Qiu et al., 2024) leverages AMR graphs to represent document structures and guide factual summarization generation, focusing on structured summarization rather than direct fact verification. Unlike previous methods, our approach integrates a trainable GNN with an LLM, combining long-form contextual understanding with structured knowledge from extracted KGs. By incorporating graph reasoning, our model captures complex entity relationships and logical structures, enabling fine-grained fact verification in a single comparison. This enhanced reasoning ability allows the model to generalize effectively to specialized domains.

3 GraphCheck

In this section, we introduce our *GraphCheck* framework, designed for efficient fact-checking. Intuitively, GraphCheck first extracts structural information from KGs to enrich the input text, then leverages a LLM for verification. GraphCheck contains of three main steps: (1) Given a source document D and a generated claim C, we extract knowledge triples from them and construct corresponding KGs. (2) A trainable GNN encodes the entire graph, generating comprehensive graph embeddings. (3) These embeddings, along with the document and claim texts, are fed into a verifier LLM, with frozen parameters, enabling single-call fine-grained fact-checking with the help of structured graph information, as shown in Figure 2.

3.1 Graph Construction

To construct the KGs, we extract triples in the form of $\{source, relation, target\}$ from the text, where each entity and relation captures key seman-

tic information. To achieve this, a LLM is employed to automatically identify and extract these triples. The detailed prompt used for triple extraction is provided in Appendix F. Building on the extracted triplets, we construct a directed graph $G = (\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V} = \{\mathbf{v}_i\}_{i=1,\ldots,n}$ is the set of node (entity) features, where each $\mathbf{v}_i$ denotes the feature vector for node i. $\mathcal{E} = \{\mathbf{e}_{ij}\}_{i,j=1,\ldots,n}$ is the set of edge (relation) features, where $\mathbf{e}_{ij}$ denotes the edge feature vector for an edge from node i to node j. The node features and edge features from textual attributes are encoded using Sentence-Transformers.[2] For a given generated claim C and its source document D we extract the corresponding graphs G_{C} and G_{D}.

3.2 GraphCheck Verification

Graph Encoding. We encode the extracted KGs with a GNN. Specifically, for the l-th GNN layer updates node features based on the message passing scheme as:

$$\mathbf{v}_i^{l+1} = \text{UPDATE}\left(\mathbf{v}_i^l, \sum_{j\in\mathcal{N}_i} \text{MESSAGE}\left(\mathbf{v}_j^l, \mathbf{e}_{ji}\right)\right),$$

where $\mathcal{N}_i$ denotes the set of node i's neighbors, and UPDATE and MESSAGE functions are implemented by neural networks. The final graph embeddings $\mathbf{h}_g$ are obtained with the GNN output layer, which is implemented with a READOUT function:

$$\mathbf{h}_{\text{g}} = \text{READOUT}\left(\{\mathbf{v}_i^L\}_{i=1,\ldots,n}\right).$$

Here, $\mathbf{v}_i^L$ indicates the feature vector of node i at the last layer. Specifically, the READOUT function includes a summation function to capture a global representation of the graph.

Text Encoding. For a given generated claim C and the source document D, we concatenate them following the verifying template shown in Appendix F, and pass the rendered prompt into the verifier LLM to obtain the text embedding $\mathbf{h}_t$.

Graph Projection. To align the graph features with the verifier LLM's textual embedding space, we employ a projector module P. This module maps the extracted graph features of claim $\mathbf{h}_{\text{g}}^{\text{C}}$ and document $\mathbf{h}_{\text{g}}^{\text{D}}$ into the LLM's embedding space, resulting in the projected graph embeddings $\tilde{\mathbf{h}}_{\text{g}}^{\text{C}}$ and $\tilde{\mathbf{h}}_{\text{g}}^{\text{C}}$ for the claim and document, respectively.

[2] https://huggingface.co/sentence-transformers/all-roberta-large-v1

Dataset	Size	Doc$_{len}$	Claim$_{len}$	Neg%
General Domain				
AggreFact-Xsum	558	324	23	48.9%
AggreFact-CNN	558	500	55	10.2%
Summeval	1600	359	63	18.4%
ExpertQA	3702	432	26	19.8%
Medical Domain				
COVID-Fact	4086	72	12	68.3%
PubHealth	1231	77	14	51.3%
SCIFact	809	249	12	58.9%

Table 1: Statistics of Benchmark Datasets. We report the size of each benchmark, the average text length of source documents and generated claims, and the proportion of negative samples.

Fact-Checking. After obtaining the projected graph embeddings, $\tilde{\mathbf{h}}_{\text{g}}^{\text{C}}$ and $\tilde{\mathbf{h}}_{\text{g}}^{\text{D}}$, along with the text embedding $\mathbf{h}_t$, we concatenate them to construct the final input representation, which is then fed into the LLM self-attention layers for fact-checking:

$$y = \text{LLM}(\tilde{\mathbf{h}}_{\text{g}}^{\text{C}}, \tilde{\mathbf{h}}_{\text{g}}^{\text{D}}, \mathbf{h}_t),$$

where $y \in \{\text{“support”}, \text{“unsupport”}\}$. The model consider both the structured and textual information to determine whether the document supports the claim.

By incorporating graph embeddings, our method effectively captures complex multi-hop logic relations in long text while ensuring efficient fact-checking. The knowledge graph, which encodes entity relationships within the entire text, assists the LLM in detecting factual inconsistencies that may be overlooked when relying solely on text.

4 Experimental Setup

4.1 Datasets

Training Dataset. To train a GNN for extracting KG information, we use the {claim, document, label} pairs from MiniCheck dataset (Tang et al., 2024) with 14K synthetic samples. We use Claude-3.5-Sonnet (Anthropic) to extract KG triples for claims and documents, constructing graphs for each pair. The final training dataset is structured as {C, D, G_{C}, G_{D}, label}. The dataset is split into training, validation, and test sets in a 6:2:2 ratio for model training and evaluation.

Evaluation Benchmarks. Our work mainly focuses on fact-checking tasks that involve long-term text, as shown in Table 1. Therefore, we adopt widely used datasets like AggreFact-CNN (Tang

Method	General Domain				Medical Domain			Overall Avg. (%)
	AggreFact -Xsum	AggreFact -CNN	Summeval	ExpertQA	COVID-Fact	SCIFact	PubHealth	
Large-scale LLMs*								
GPT-4 (Achiam et al., 2023)	75.4	60.7	69.7	59.6	73.8	83.3	73.2	70.8
GPT-4o (Hurst et al., 2024)	76.4	66.8	76.3	58.3	62.6	83.2	67.0	70.1
OpenAI o1 (Jaech et al., 2024)	74.8	65.3	70.5	58.8	75.9	90.3	74.8	72.9
Claude 3.5-Sonnet (Anthropic)	75.7	68.8	77.3	58.8	73.8	87.2	73.8	73.6
DeepSeek-V3 671B(Liu et al., 2024a)	74.6	63.2	68.3	58.5	75.9	89.1	72.9	71.7
Small-scale LLMs								
Llama3 8B (Dubey et al., 2024)	53.4	51.3	51.7	51.3	58.1	62.2	70.7	57.0
Qwen2.5 7B (Yang et al., 2024a)	53.2	45.3	58.5	53.6	59.2	53.5	59.1	54.7
Llama3.3 70B (Dubey et al., 2024)	60.1	53.5	57.6	54.3	69.0	85.7	76.9	65.3
Qwen2.5 72B (Yang et al., 2024a)	55.6	49.9	53.4	54.1	69.9	85.6	76.7	63.6
Specialized Fact-checking Methods								
AlignScore (Zha et al., 2023)	68.0	54.1	62.2	59.3	66.5	71.7	64.4	63.7
ACUEval (Wan et al., 2024)	55.5	50.0	53.7	57.5	64.7	79.9	62.9	60.6
MiniCheck (Tang et al., 2024)	70.8	63.7	74.8	57.4	65.9	78.1	66.3	68.1
GraphEval (Sansford et al., 2024)	67.6	69.5	69.7	56.0	60.7	68.4	63.7	65.1
Ours								
GraphCheck-Llama3.3 70B	72.9	62.4	67.3	60.3	71.5	89.4	73.6	71.1
GraphCheck-Qwen 72B	72.1	66.5	71.0	57.2	69.7	86.4	71.7	70.7

Table 2: Balanced accuracy of fact-checkers across all benchmarks, covering both general and medical domains. Methods are categorized into ***Large-scale LLMs**** | ***Small-scale LLMs*** | ***Specialized Fact-checking Methods*** | ***Ours***. The top-1 , top-2 , and top-3 performances for each dataset among models smaller than Large-scale LLMs are highlighted, while the best-performing results within Large-scale LLMs are underlined.

et al., 2023), AggreFact-XSum (Tang et al., 2023), and SummEval (Fabbri et al., 2021), all of which include lengthy documents. To assess our method's performance in open-domain scenarios, we also incorporate the long-text question-answering dataset ExpertQA (Malaviya et al., 2023). Furthermore, we evaluate our method on medical datasets, including SciFact (Wadden et al., 2020), COVID-Fact (Saakyan et al., 2021), and PubHealth (Kotonya and Toni, 2020), which provide specialized medical domain information. More details are shown in Appendix A.

4.2 Baselines

To comprehensively evaluate our method, we compare it against various fact-checkers, categorized into large-scale LLMs, small-scale LLMs, and specialized fact-checking methods.

Large-scale LLMs[3] include GPT-4 (Achiam et al., 2023), GPT-4o (Hurst et al., 2024), OpenAI o1 (Jaech et al., 2024), Claude 3.5-Sonnet (Anthropic), and the largest open source model DeepSeek-V3 671B (Liu et al., 2024a). For small-scale LLMs, we include Llama3 8B, Llama3.3 70B (Dubey et al., 2024), Qwen2.5 7B, and Qwen2.5 72B (Yang et al., 2024a). For specialized fact-checking methods, we include AlignScore (Zha et al., 2023) and fine-grained fact-checkers like MiniCheck (Tang et al., 2024) and ACUEval (Wan et al., 2024). Additionally, we also consider graph-based methods, namely GraphEval (Sansford et al., 2024) and GraphRAG (Edge et al., 2024).

[3]We consider Large-scale LLMs as models with more than 300 B parameters.

4.3 Evaluation Metric

Considering the data imbalance in some benchmarks, models biased towards a particular class in predictions may not reflect their true performance. To address this, we follow previous approached (Liu et al., 2023b; Tang et al., 2023) and calculate balanced accuracy (BAcc). For more implementation details, please refer to Appendix D.

5 Results and Analysis

5.1 Main Results

Table 2 presents the BAcc of our proposed method, GraphCheck, compared to LLMs and specialized fact-checkers across general and medical domain benchmarks. The results show that our proposed GraphCheck achieves strong performance, reaching an overall BAcc of 71.1% across all benchmarks. Specifically, among large-scale LLMs, Claude 3.5-Sonnet achieves the best overall performance. Our method outperforms GPT-4 and GPT-4o and comes close to the most advanced large-scale models, including OpenAI o1, Claude 3.5-Sonnet, and the latest open-source model DeepSeek-V3 671B, while operating at a smaller scale and significantly lower cost. Interestingly,

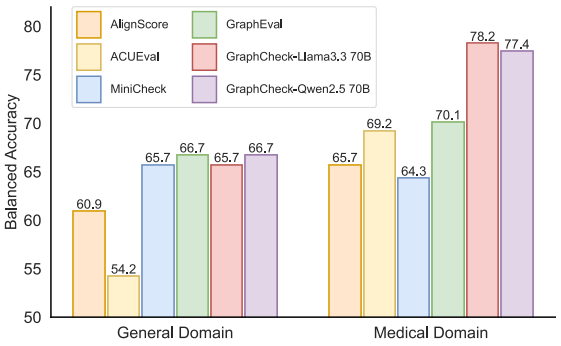

Figure 3: Average BAcc across general and medical domains. We compare our method with the specialized fact-checking methods in general domain (AggreFact-XSum, AggreFact-CNN, Summeval, ExpertQA) and medical domain (COVID-Fact, PubHealth, SCIFact).

GPT-4o underperforms on the medical datasets COVID-Fact and PubHealth, which contain shorter texts, even scoring lower than GPT-4. For small-scale LLMs, our method achieves a 5.8% and 7.5% improvement over its similarly sized models, Llama3.3 70B and Qwen2.5 72B, respectively. For Specialized Fact-checking Methods, GraphCheck outperforms all methods, achieving 10.5%, 3%, and 6% improvements over ACUEval, MiniCheck, and GraphEval, respectively. Notably, compared to methods that require multiple calls, our method achieves superior performance with a single model call. Although GraphRAG is not typically used for fact-checking, its popularity motivated us to adapt it for this purpose. A detailed analysis of these adaptations is provided in the Appendix G.

In particular, our method achieves a BAcc of 60.3% on ExpertQA, surpassing all models. This may be because GraphCheck can extract complex logical relations from graph data. However, our method underperforms on AggreFact-CNN and Summeval, which contain longer claims (average length > 50) and include more factual details. This makes knowledge triplets extraction more challenging, as some important information may be lost during the process, affecting subsequent fact-checking.

Performance Analysis in Different Domains. To evaluate the effectiveness of our method across different domains, we compare it with other specialized fact-checking methods in both general and medical domains, as shown in Figure 3. In the general domain, our method matches the performance of approaches like MiniCheck and GraphEval, which require multiple calls. However, in the medical domain, our method significantly outperforms these methods, achieving an 8.1% improvement over Minicheck. This demonstrates the strong generalization ability of our method, when other methods perform limited in the medical domain, our method still maintains strong performance.

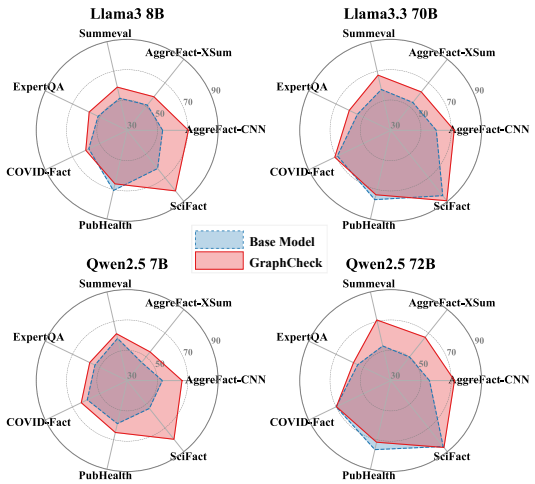

Figure 4: The BAcc of the base model and the proposed GraphCheck architecture across all seven benchmarks for Llama3 8B, Llama3.3 70B, Qwen2.5 7B, Qwen2.5 72B models. The blue-shaded region represents the base model performance, while the red-shaded region highlights the enhanced performance with GraphCheck.

5.2 Ablation Studies

Impact of Additional Graph Information. To evaluate the effectiveness of we proposed incorporating graph information, we compare (1) the base LLM models with (2) our proposed GraphCheck, which is based on these models. As shown in Figure 4, our approach has a significant improvement on both lightweight models (Llama3 8B[4], Qwen2.5 7B[5]) and larger models (Llama3.3 70B, Qwen2.5 72B). Specifically, as shown in Table 1, our method achieves significant improvement on relatively long-text datasets AggreFact-XSum, AggreFact-CNN, and Summeval. In contrast, for shorter-text datasets like COVID-Fact, the improvement is minimal, especially on PubHealth, our method even shows a slight performance drop. This is because the base models already handle fact-checking effectively on shorter, simpler texts, and introducing additional graph information may interfere with their inference.

A similar result is observed on the relatively longer SCIFact dataset, where our approach significantly enhances lightweight models. However, for larger models, which can already handle longer texts effectively, the improvement is much more

[4] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[5] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

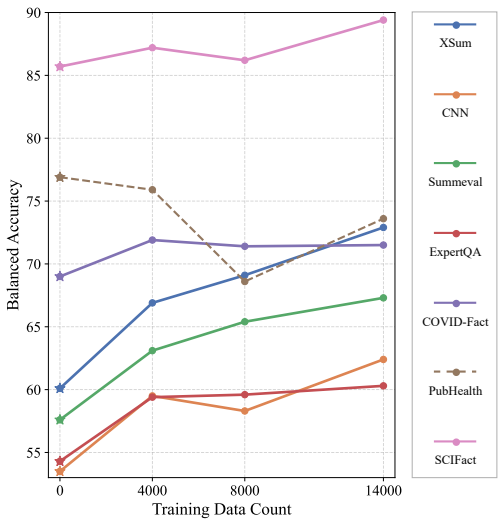

Figure 5: Balanced accuracy comparison across different training data sizes on all benchmarks. The baseline model performance is marked at 0 on the x-axis.

limited. The above results demonstrate the effectiveness of our method, showing that GraphCheck enhances the ability of models to handle long-text fact-checking tasks. Additionally, graph information is essential for effectively capturing complex logical relations within the text.

Impact of Training Data Sizes. To evaluate the impact of training data size on the fact-checking performance of GraphCheck-Llama3.3 70B, we conducted experiments across all benchmarks, as shown in Figure 5. The results demonstrate a general upward trend in model performance as the amount of training data increases. Specifically, significant improvements can be observed on the long-text datasets AggreFact-XSum, AggreFact-CNN, Summeval, and SCIFact. Among them, XSum exhibits the largest improvement, increasing from 60.1% to 72.9%, while CNN and Summeval also achieve approximately 10% improvements. In contrast, for the short-text datasets, our method shows only a slight improvement on COVID-Fact, while on PubHealth, performance gradually declines. These results further validate the conclusion drawn in Section 5.2

From the above results, we can observe that as the training data size increases, the overall model performance shows a upward trend. Therefore, we believe that further increasing the data size could continue to enhance the performance of our proposed GraphCheck.

Impact of Generated Knowledge Graph Quality. Due to the inherent randomness in extracting entity-relationship triples from text using LLMs, we conducted an experiment to assess how the quality of KGs generated from text impacts the model's final fact-checking results, as illustrated in Figure 6. For shorter generated claims, the triples extracted by the four models show minimal differences, except for occasional missing details (e.g., "Saturday") by GPT-4o and DeepSeek-V3. These missing have minor effects on fact-checking results, as the models also relied on the original text for verification. In contrast, for longer document texts, there are significant differences in the quality of the triples generated by the models. Specifically, the triplets extracted by the Llama 8B model lacked crucial details, such as the time ("Saturday") and the reason for the exam cancellation. The loss of key information could turn the KG into interference during fact-checking, ultimately leading to incorrect results. On the other hand, while the language expressions of the triples extracted by GPT-4, Claude 3.5, and DeepSeek-V3 are different, they all capture the essential details and still ensure the fact-checker makes the right prediction.

These findings indicate that for short texts, the quality of the extracted KG has minimal impact on fact-checking performance, as models still rely on the original text for verification. However, for long-form documents, the completeness of the KG is critical. If the KG lacks key fact-checking information, it misleads the model rather than assists in verification. This is because longer texts make it more difficult for the model to extract essential details directly, increase its dependence on the KG. In such cases, an incomplete or inaccurate KG introduces noise and ultimately compromises fact-checking accuracy. Conversely, if the missing information is irrelevant to the verification process, its absence does not affect the result.

5.3 Rethink Graph Importance on Long-Form Fact-checking

We conduct a case study in the medical domain to demonstrate how our method uses KGs to help LLMs in the fact-checking process. We also showcase how our approach provides clear and interpretable explanations for the final checking results, as shown in Figure 7. For each edge in the graph, we visualize its connection weight to show the attention distribution learned by the GNN model. The results indicate that the model selectively focuses on specific edges by assigning higher attention weights, emphasizing key relationships in the graph. Notably, these high-weight triplets cor-

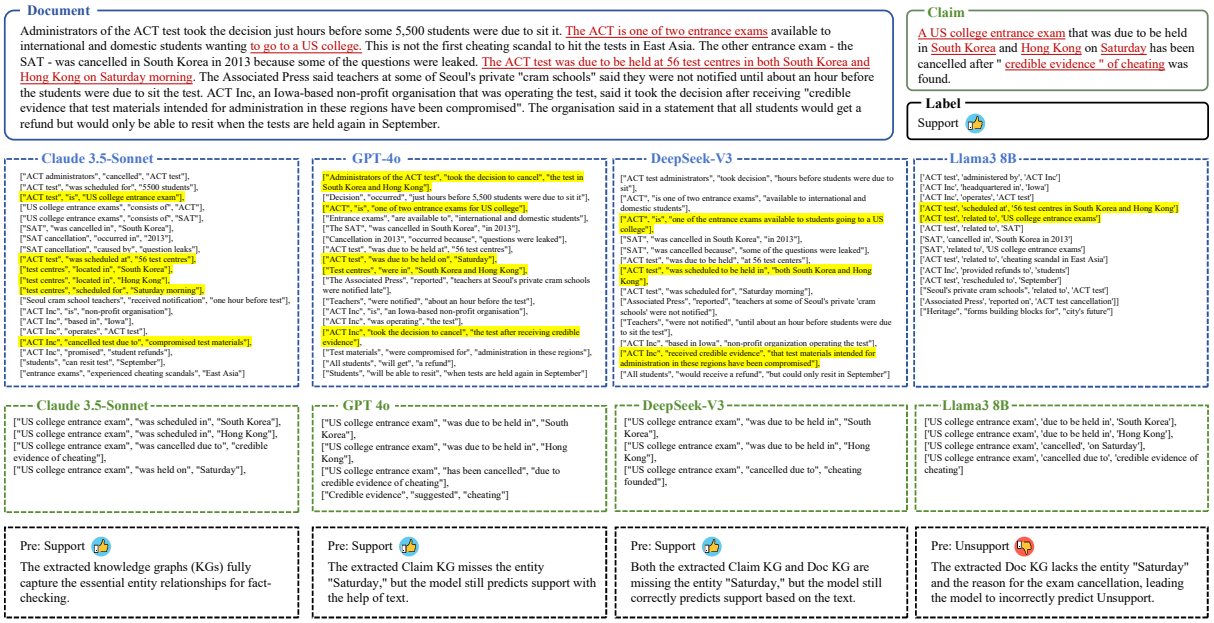

Figure 6: Example Analysis of the Impact of Knowledge Graph (KG) Quality on Model Prediction Results. The figure illustrates the influence of KGs extracted by four different models (Claude 3.5-Sonnet, GPT-4o, DeepSeek-V3, Llama 8B) on the performance of GraphCheck fact-checking.

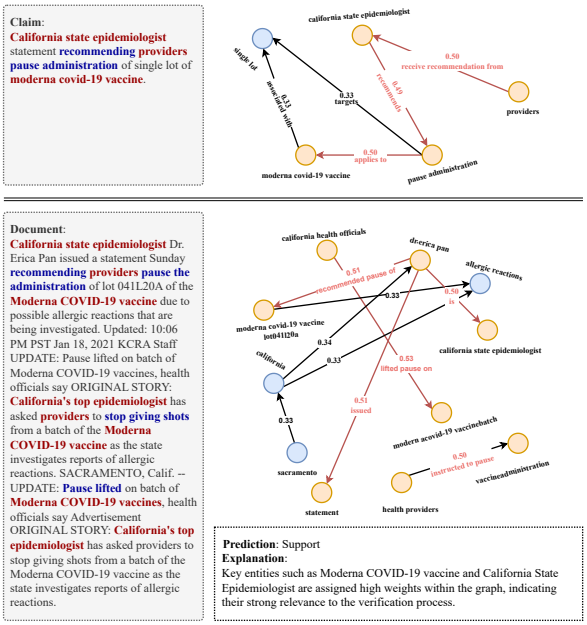

Figure 7: A case study in the medical domain. Connection weights in the KG are visualized to highlight key relationships primarily used by the model for fact-checking. Key entities and relationships in the text are marked in red and blue, while high-weighted nodes in the KG are highlighted in orange-yellow.

respond to key relations that align with the fact-checking requirements. For instance, the triplets (Dr. Erica Pan, is, California state epidemiologist) and (Dr. Erica Pan, recommended pause of, Moderna COVID-19 vaccine) in the document KG capture key information needed to verify the claim.

Explainability. This visualization not only highlights the key information the model relies on, but also improves the explainability of its fact-checking process. By revealing which relationships receive higher attention, it becomes easier to understand how the model makes its final decision and incorporates graph reasoning into its predictions. This explainability is particularly important in the medical domain, where fact-checking requires a clear and reliable reasoning path.

6 Conclusion

In this work, we propose GraphCheck, a fact-checking method that integrates knowledge graphs to enhance LLM-based fact-checking, particularly for long-form text. GraphCheck addresses the limitations of LLMs in capturing complex entity relationships, which often result in overlooked factual errors. By leveraging graph neural networks (GNNs) to integrate representations from the generated claim and the source document KGs, our method enables fine-grained fact-checking in a single model call, significantly improving efficiency. Furthermore, the incorporation of graph information enhances the interpretability of the fact-checking process. Experiments on general and medical domain datasets demonstrate that GraphCheck achieves competitive performance.

Limitations

Quality of Knowledge Graphs. Although integrating KGs into the fact-checking process is effective, our method remains limited by the quality of KGs. Currently, there is no reliable method for evaluating the quality of extracted KGs, and the process largely depends on manual judgment. As the dataset grows, it becomes difficult to assess the quality of the extracted KGs. As analyzed in our paper, KG quality directly impacts our method's performance (errors in the KG may introduce noise or fail to provide sufficient support for fact-checking).
Training Data. Our method was trained on a 14k dataset with relatively short text. As shown in the paper, performance improves as the training data increases, indicating that there is still potential for further improvement if larger or higher-quality datasets are considered.

References

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haim ing Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Made laine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Benjamin Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Sim'on Posada Fishman, Juston Forte, Is abella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Raphael Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Jo hannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Lukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Ryan Kiros, Matthew Knight, Daniel Kokotajlo, Lukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Ma teusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel P. Mossing, Tong Mu, Mira Murati, Oleg Murk, David M'ely, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Ouyang Long, Cullen O'Keefe, Jakub W. Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alexandre Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Pondé de Oliveira Pinto, Michael Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack W. Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario D. Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas A. Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cer'on Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll L. Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qim ing Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Hiba Ahsan, Denis Jered McInerney, Jisoo Kim, Christopher Potter, Geoffrey Young, Silvio Amir, and Byron C. Wallace. 2023. Retrieving evidence from ehrs with llms: Possibilities and challenges. *Proceedings of machine learning research*, 248:489–505.

Anthropic. Claude 3.5 sonnet. https://www.anthropic.com/news/claude-3-5-sonnet. Retrieved February 12, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela

Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Wenqi Fan, Yujuan Ding, Liang bo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Knowledge Discovery and Data Mining*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Fan Gao, Hang Jiang, Rui Yang, Qingcheng Zeng, Jinghui Lu, Moritz Blum, Dairui Liu, Tianwei She, Yuang Jiang, and Irene Li. 2023. Evaluating large language models on wikipedia-style survey generation. In *Annual Meeting of the Association for Computational Linguistics*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Hang Jiang, Xiajie Zhang, Robert Mahari, Daniel Kessler, Eric Ma, Tal August, Irene Li, Alex 'Sandy' Pentland, Yoon Kim, Jad Kabbara, and Deb Roy. 2024. Leveraging large language models for learning complex legal concepts through storytelling. *ArXiv*, abs/2402.17019.

Jiho Kim, Sungjin Park, Yeonsu Kwon, Yohan Jo, James Thorne, and Edward Choi. 2023. Factkg: Fact verification via reasoning on knowledge graphs. *arXiv preprint arXiv:2305.06590*.

Kyungha Kim, Sangyun Lee, Kung-Hsiang Huang, Hou Pong Chan, Manling Li, and Heng Ji. 2024. Can llms produce faithful explanations for fact-checking? towards faithful explainable fact-checking via multi-agent debate. *ArXiv*, abs/2402.07401.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. *arXiv preprint arXiv:2010.09926*.

Jean Lee, Nicholas Stevens, Soyeon Caren Han, and Minseok Song. 2024. A survey of large language models in finance (finllms). *ArXiv*, abs/2402.02315.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Dairui Liu, Boming Yang, Honghui Du, Derek Greene, Aonghus Lawlor, Ruihai Dong, and Irene Li. 2023a. Recprompt: A prompt tuning framework for news recommendation using large language models. *arXiv preprint arXiv:2312.10463*.

Xiaoze Liu, Feijie Wu, Tianyang Xu, Zhuo Chen, Yichi Zhang, Xiaoqian Wang, and Jing Gao. 2024b. Evaluating the factuality of large language models using large-scale knowledge graphs. *arXiv preprint arXiv:2404.00942*.

Yinhong Liu, Zhijiang Guo, Tianya Liang, Ehsan Shareghi, Ivan Vulić, and Nigel Collier. 2024c. Aligning with logic: Measuring, evaluating and improving logical consistency in large language models. *arXiv preprint arXiv:2410.02205*.

Yixin Liu, Alexander Fabbri, Yilun Zhao, Pengfei Liu, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023b. Towards interpretable and efficient automatic reference-based summarization evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16360–16368, Singapore. Association for Computational Linguistics.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2023. Expertqa: Expert-curated questions and attributed answers. *arXiv preprint arXiv:2309.07852*.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages

2758–2774, Singapore. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *arXiv preprint arXiv:2305.15852*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Haoyi Qiu, Kung-Hsiang Huang, Jingnong Qu, and Nanyun Peng. 2024. AMRFact: Enhancing Summarization Factuality Evaluation with AMR-Driven Negative Samples Generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 594–608, Mexico City, Mexico. Association for Computational Linguistics.

Sanjana Ramprasad, Elisa Ferracane, and Zachary C Lipton. 2024. Analyzing llm behavior in dialogue summarization: Unveiling circumstantial hallucination trends. *arXiv preprint arXiv:2406.03487*.

Leonardo FR Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, and Mohit Bansal. 2022. Factgraph: Evaluating factuality in summarization with semantic graph representations. *arXiv preprint arXiv:2204.06508*.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. Covid-fact: Fact extraction and verification of real-world claims on covid-19 pandemic. *arXiv preprint arXiv:2106.03794*.

Hannah Sansford, Nicholas Richardson, Hermina Petric Maretic, and Juba Nait Saada. 2024. Grapheval: A knowledge-graph based llm hallucination evaluation framework. *arXiv preprint arXiv:2407.10793*.

Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryscinski, Justin Rousseau, and Greg Durrett. 2023. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11626–11644, Toronto, Canada. Association for Computational Linguistics.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. Minicheck: Efficient fact-checking of llms on grounding documents. *arXiv preprint arXiv:2404.10774*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

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

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. *arXiv preprint arXiv:2004.14974*.

David Wan, Koustuv Sinha, Srini Iyer, Asli Celikyilmaz, Mohit Bansal, and Ramakanth Pasunuru. 2024. Acueval: Fine-grained hallucination evaluation and correction for abstractive summarization. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10036–10056.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Rui Yang, Haoran Liu, Edison Marrese-Taylor, Qingcheng Zeng, Yuhe Ke, Wanxin Li, Lechao Cheng, Qingyu Chen, James Caverlee, Yutaka Matsuo, and Irene Li. 2024b. KG-rank: Enhancing large language models for medical QA with knowledge graphs and ranking techniques. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 155–166, Bangkok, Thailand. Association for Computational Linguistics.

Rui Yang, Ting Fang Tan, Wei Lu, Arun James Thirunavukarasu, Daniel Shu Wei Ting, and Nan Liu. 2023. Large language models in health care: Development, applications, and challenges. *Health Care Science*, 2(4):255–263.

Zhangdie Yuan and Andreas Vlachos. 2023. Zero-shot fact-checking with semantic triples and knowledge graphs. *arXiv preprint arXiv:2312.11785*.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. *Advances in neural information processing systems*, 32.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. *arXiv preprint arXiv:2305.16739*.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

A Benchmark Details

A.1 General Domain Benchmarks

AggreFact-XSum, AggreFact-CNN. They are subsets of the AGGREFACT benchmark (Tang et al., 2023), designed for evaluating factual consistency in summarization. These subsets correspond to two widely used summarization datasets: XSum (Nallapati et al., 2016) and CNN/DailyMail (CNN/DM) (Narayan et al., 2018), which feature different summarization styles. Both datasets contain relatively long documents, making them well-suited for assessing our method's effectiveness in handling long-text fact-checking.

Summeval. (Fabbri et al., 2021) consists of human evaluations of 16 summarization model outputs based on 100 articles from the CNN/DailyMail dataset. Each summary is rated on a Likert scale from 1 to 5 across four categories: consistency, coherence, fluency, and relevance. In our use of this dataset, we extract each individual claim from the summaries as separate data points. The consistency score is mapped such that a score of 5 is labeled as Support, while scores ranging from 0 to 4 are labeled as Unsupport.

ExpertQA. (Malaviya et al., 2023) includes responses from six different systems to expert-curated queries, with sentence-level verification against cited or retrieved documents. In our dataset, a sentence is labeled as Support only if the evidence fully supports it. In contrast, partial and incomplete support is classified as Unsupport.

A.2 Medical Domain Benchmarks

COVID-Fact. (Saakyan et al., 2021) is a dataset containing 4,086 claims related to the COVID-19 pandemic. The dataset focuses on automatically detecting true claims and their corresponding source articles, followed by generating counter-claims using automated methods instead of human annotators.

PubHealth. (Kotonya and Toni, 2020) consists of 11,832 claims related to a variety of health topics, including biomedical subjects such as infectious diseases and stem cell research, government healthcare policies like abortion, mental health, and women's health, as well as other public health-related issues. Each claim in the dataset is paired with journalist-crafted, gold-standard explanations that provide judgments to support the corresponding fact-check labels. The dataset is designed for two main tasks: veracity prediction and explanation generation, with claims categorized into four labels: true, false, mixture, and unproven. In our experiments, we use the test set as a benchmark, classifying claims labeled as true as Support, while those labeled as false, mixture, and unproven are classified as Unsupport.

SCIFact. (Wadden et al., 2020) consists of 1,400 expert-written scientific claims, each paired with evidence-containing abstracts annotated with labels and rationales. To construct the dataset, annotators re-formulate naturally occurring claims found in scientific literature—specifically citation sentences—into atomic scientific claims, ensuring clarity and precision. Since its training set is labeled, we use it as a benchmark in our experiments. Furthermore, claims with contradictory evidence or no supporting evidence are classified as Unsupport, while all others are classified as Support.

A.3 Preprocessing for Benchmark

To extract graph information from the benchmark text data, we utilize LLM to separately extract entity-relation triples from both the claims and the documents. The extraction process follows the prompt shown in F. After preprocessing, the dataset is structured as {claim, doc, claim_kg, doc_kg, label}, where claim_kg and doc_kg represent the extracted KGs for the claim and document, respectively. Samples of the processed data are illustrated in Figure 8.

B Synthetic Dataset for Training

To pre-train an external GNN, we synthesized a structured dataset of 14,000 samples based on the MiniCheck training set[6]. Using a method similar to A.3, we employed the Qwen2.5 7B model to extract KG triples from both the claim and document in each sample, following the prompt in F. Each sample is structured as {claim, doc, claim_kg, doc_kg, label}. Examples are shown in Figure 9.

[6]We will release upon acceptance.

Model	Avg. Calls per Sample	Inference time per Sample (secs)	Cost ($)
GPT-4	1	7.1	18.6
OpenAI o1	1	17.4	27.7
Claude 3.5-Sonnet	1	8.2	7.2
MiniCheck	5	0.01	< 1.0
ACUEval	5	5.9	8.8
GraphEval	9	0.51	< 1.0
GraphCheck(Ours)	1	0.68	<1.0

Table 3: Comparison of the cost of our method with other specialized fact-checking methods and LLMs.

C Analysis of Computational Cost and Time Efficiency

We compare the computational cost of specialized fact-checking methods and LLMs on the ExpertQA benchmark, selected for its large dataset size and longer text length. For locally deployed methods, we calculate the cost at a rate of $0.8 per GPU hour, as shown in Table 3. The results show that the cost of our method is significantly lower than that of similar LLMs, such as GPT-4, OpenAI O1, and Claude 3.5-Sonnet. Compared to specialized fact-checking methods, our approach shows a substantial efficiency improvement over ACUEval, which also uses the Llama3.3 70B as the base model. Additionally, the cost of our method is comparable to that of Minicheck and GraphEval, which are based on smaller NLI models. Notably, due to the small size of NLI models, their inference speed is fast, allowing Minicheck and GraphEval to maintain low computational costs. However, this also limits their performance and generalization ability. In contrast, our approach remains computationally efficient while achieving superior performance on complex verification tasks. Specifically, our method outperforms Minicheck and GraphEval on the ExpertQA benchmark, demonstrating stronger generalization in handling long-form text scenarios.

Graph_Building Method	XSum	CNN	Summeval	ExpertQA
Edge as Input (used)	72.9	60.3	66.2	60,3
Edge as Node	72.5	59.6	66.8	58.6

Table 4: Balanced accuracy comparison of different graph building methods on XSum, CNN, Summeval and ExpertQA benchmarks.

D Implementation Details

For training, we use Llama3.3 70B [7] and Qwen2.5 72B [8] as the base models, which remain frozen throughout the training process. The trainable external graph encoder is a GNN. We train the models for 20 epochs with early stopping, setting the maximum generation length to 5 and the learning rate to 1×10^{-5}. The best model is selected based on performance on the validation set. The experiments are conducted on 4 NVIDIA A100 80GB GPUs for both training and testing.

Detail of Hyperparameter. We list all the parameters used for both Llama3.3 70B and Qwen2.5 72B models, as shown in table 5. This includes configuration details such as batch size, learning rate, and optimizer settings.

E Additional Experiments

Analyzing the Impact of Different Graph-Building Methods. We explored two different graph-building methods to evaluate the impact of graph building methods on our approach. The first method directly encodes the relation as edge information in the triplet, represented as [entity1, relation, entity2]. The second method treats the relation as a node, represented as [entity1 → relation] and [relation → entity2].

[7] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct

[8] https://huggingface.co/Qwen/Qwen2.5-72B-Instruct

Model	XSum	CNN	Summeval	ExpertQA
Llama3.3 70B	60.1	53.5	57.6	54.3
Llama3.3 70B + GAT	72.9	59.6	65.4	60.3
Llama3.3 70B + GT	64.8	62.4	67.3	59.1

Table 5: Balanced accuracy comparison of different GNN architectures on XSum, CNN, Summeval and ExpertQA benchmarks.

As shown in Table 4, the results show that directly encoding edge information leads to slightly better performance compared to treating the relation as a node, although the difference is minimal.

Method	**AggreFact -Xsum**	**AggreFact -CNN**	**Summeval**	**ExpertQA**	**COVID-Fact**	**SCIFact**	**PubHealth**	**Overall** Avg. (%)
GraphRAG (GPT-4o) (Edge et al., 2024)	54.2	57.9	51.5	51.8	50.9	59.1	52.5	54.0
GraphCheck-Llama3.3 70B (Ours)	72.9	62.4	67.3	60.3	71.5	89.4	73.6	71.1
GraphCheck-Qwen 72B (Ours)	72.1	66.5	71.0	57.2	69.7	86.4	71.7	70.7

Table 6: Balanced accuracy of GraphRAG and GraphCheck across all evaluation benchmarks.

Impact of Different GNN Architecture. In our study, we explore the effect of different GNN architectures—Graph Attention Network (GAT) (Veličković et al., 2017) and Graph Transformer (GT) (Yun et al., 2019). As shown in Table 5, the experimental results demonstrate that for the XSum dataset, GAT significantly improves performance from the baseline of 60.1 to 72.9% (+12.8%), while GT achieves a smaller improvement of 4.7% (64.8%). This suggests that XSum relies more on local relationships, where GAT's attention mechanism effectively captures interactions between adjacent nodes. In contrast, GT's global self-attention may introduce noise or lead to over-smoothing, limiting its effectiveness. However, for Summeval and CNN, GT outperforms GAT (Summeval: 67.3% vs. 65.4%, CNN: 62.4% vs. 59.6%), suggesting that tasks requiring long-range dependencies and global context benefit more from GT's ability to integrate information across the graph structure. For the ExpertQA dataset, both GAT and GT exhibit similar performance.

F Prompts

Triplets Extraction. Figure 10 presents the prompt and an example used for extracting entity-relation triples from a text using an LLM. The example is sourced from the COVID-Fact dataset.
Fact-Checking. Figure 11 presents the fact-checking prompt and an example output. Compared to our method (first row), zero-shot LLMs (second row) require additional descriptive instructions to ensure the stability of the generated output format.

G GraphRAG Evaluation

Implementation Details of GraphRAG To streamline our implementation process, we leveraged the approach from the open-source nano-GraphRAG project[9] for our testing phase. During our trials, we employed the OpenAI text-embedding-3-small model for embeddings and the updated GPT-4o model for inference, yielding outputs indicating support or unsupported. The operational workflow is divided into two phases: **Insert** and **Query**.

In the **Insert** phase, the input document is segmented into smaller text blocks. Each block undergoes entity extraction to construct a text block-entity relationship graph, which is then clustered to form community reports.

In the **Query** phase, semantic similarity is used to retrieve relevant information from the document KG, aggregating the community reports associated with the retrieved text blocks. Finally, responses are generated using an LLM based on the retrieved context. The prompt used for response generation is as follows:

[9] https://github.com/gusye1234/nano-graphrag

Hyperparameter	Value
batch_size	8
num_epochs	20
learning_rate	1e-5
weight_decay	0.05
warmup_epochs	2
early_stop_patience	3
llm_num_virtual_tokens	4
max_txt_len	1024
max_new_tokens	5, 8
gnn_model	gat, gt
gnn_num_layers	2, 3, 4
gnn_in_dim	1024
gnn_hidden_dim	1024
gnn_num_heads	4
gnn_dropout	0.3, 0.4, 0.5

Table 7: Hyperparameters.

Based on the KG content, evaluate if the claim triplets are fully supported.

Claim to verify: {claim kg}

Instructions:

- Respond ONLY with a single digit.

- Return '1' if ALL claim triplets are fully supported by the graph.

- Return '0' if ANY claim triplet is not supported or only partially supported.

We compare GraphRAG and GraphCheck in Table 6, where GraphRAG generally underperforms. In our experiments, the LLM often defaulted to producing "0" outputs, leading to suboptimal GraphRAG scores. This behavior may stem from our simplified GraphRAG implementation, which could introduce biases through lower-precision embedding and retrieval stages. The LLM's inclination toward "0" outputs also highlights a need for more nuanced prompt design. Since GraphRAG is a general-purpose framework rather than one specifically tailored for fact-checking, we suggest it may not be an ideal approach for this task.

Claim	Doc	Claim_kg	Doc_kg	Label
donald sterling , nba team last year . sterling 's wife sued for $ 2.6 million in gifts . sterling says he is the former female companion who has lost the . sterling has ordered v. stiviano to pay back $ 2.6 m in gifts after his wife sued . sterling also includes a $ 391 easter bunny costume , $ 299 and a $ 299 .	(CNN) Donald Sterling's racist remarks cost him an NBA team last year. But now it's his former female companion who has lost big. A Los Angeles judge has ordered V. Stiviano to pay back more than $2.6 million in gifts after Sterling's wife sued her. In the lawsuit, Rochelle "Shelly" Sterling accused Stiviano of targeting extremely wealthy older men. She claimed Donald Sterling used the couple's money to buy Stiviano a Ferrari, two Bentleys and a Range Rover, and that he helped her get a $1.8 million duplex. Who is V. Stiviano? Stiviano countered that there was nothing wrong with Donald Sterling giving her gifts and that she never took advantage of the former Los Angeles Clippers owner, who made much of his fortune in real estate. Shelly Sterling was thrilled with the court decision Tuesday, her lawyer told CNN affiliate KABC. "This is a victory for the Sterling family in recovering the $2,630,000 that Donald lavished on a conniving mistress," attorney Pierce O'Donnell said in a statement. "It also sets a precedent that the injured spouse can recover damages from the recipient of these ill-begotten gifts." Stiviano's gifts from Donald Sterling didn't just include uber-expensive items like luxury cars. According to the Los Angeles Times, the list also includes a $391 Easter bunny costume, a $299 two-speed blender and a $12 lace thong. Donald Sterling's downfall came after an audio recording surfaced of the octogenarian arguing with Stiviano. In the tape, Sterling chastises Stiviano for posting pictures on social media of her posing with African-Americans, including basketball legend Magic Johnson. "In your lousy f**ing Instagrams, you don't have to have yourself with -- walking with black people," Sterling said in the audio first posted by TMZ. He also tells Stiviano not to bring Johnson to Clippers games and not to post photos with the Hall of Famer so Sterling's friends can see. "Admire him, bring him here, feed him, f**k him, but don't put (Magic) on an Instagram for the world to have to see so they have to call me," Sterling said. NBA Commissioner Adam Silver banned Sterling from the league, fined him $2.5 million and pushed through a charge to terminate all of his ownership rights in the franchise. Fact check: Donald Sterling's claims vs. reality CNN's Dottie Evans contributed to this report.	[('Donald Sterling', 'owned', 'NBA team'), ("Sterling's wife", 'sued for', '$2.6 million in gifts'), ('V. Stiviano', 'ordered to pay', '$2.6 million'), ('V. Stiviano', 'received', 'easter bunny costume'), ('easter bunny costume', 'valued at', '$391'), ("Sterling's wife", 'sued', 'V. Stiviano'), ('V. Stiviano', 'was', 'former female companion'), ('V. Stiviano', 'must return', 'gifts')]	[('Donald Sterling', 'lost', 'NBA team'), ('Donald Sterling', 'made racist remarks', 'in audio recording'), ('V. Stiviano', 'ordered to pay back', '$2.6 million in gifts'), ('Los Angeles judge', 'ordered', 'V. Stiviano repayment'), ("Rochelle 'Shelly' Sterling", 'sued', 'V. Stiviano'), ("Rochelle 'Shelly' Sterling", 'accused', 'V. Stiviano of targeting wealthy older men'), ('Donald Sterling', 'bought for', 'V. Stiviano'), ('Donald Sterling', 'purchased', 'Ferrari'), ('Donald Sterling', 'purchased', 'two Bentleys'), ('Donald Sterling', 'purchased', 'Range Rover'), ('Donald Sterling', 'helped acquire', '$1.8 million duplex'), ('V. Stiviano', 'claimed', 'gifts were legitimate'), ('Donald Sterling', 'was', 'former Los Angeles Clippers owner'), ('Donald Sterling', 'made fortune in', 'real estate'), ("Pierce O'Donnell", 'is', "Shelly Sterling's lawyer"), ('Donald Sterling', 'gave', '$391 Easter bunny costume'), ('Donald Sterling', 'gave', '$299 two-speed blender'), ('Donald Sterling', 'gave', '$12 lace thong'), ('Donald Sterling', 'argued with', 'V. Stiviano in audio recording'), ('Donald Sterling', 'criticized', "Stiviano's social media posts with African-Americans"), ('Donald Sterling', 'mentioned', 'Magic Johnson in recording'), ('Adam Silver', 'is', 'NBA Commissioner'), ('Adam Silver', 'banned', 'Donald Sterling from NBA'), ('Adam Silver', 'fined', 'Donald Sterling $2.5 million'), ('Adam Silver', 'terminated', "Sterling's ownership rights"), ('Donald Sterling', 'told Stiviano not to bring', 'Magic Johnson to games'), ('Donald Sterling', 'told Stiviano not to post photos with', 'Magic Johnson'), ('Magic Johnson', 'is', 'basketball legend'), ('Magic Johnson', 'is', 'Hall of Famer')]	0
According to the web search results, the recommended first line of treatment for HER2+ metastatic breast cancer is **chemotherapy plus HER2-directed therapy**, which are drugs that target the HER2 protein on cancer cells.	By Wade Smith, MD, as told to Kara Mayer Robinson A diagnosis of HER2-positive breast cancer can be frightening at first, especially when you hear the words "aggressive cancer." But there's reason to be optimistic about today's advances in treatment. There isn't a one-size-fits-all approach, but with the help of your doctors, you can choose what's best for you. Your Treatment Is Unique HER2-positive breast cancer is different from other breast cancer types, so your treatment won't necessarily be the same as someone else who has a different form of breast cancer. It may also be different than another HER2-positive patient's therapy. Each cancer is unique, so doctors try to develop the treatment course that's best for you. Things to consider include the size of your tumor, whether the cancer has metastasized (spread), or your overall risk of recurrence. Treatments You May Consider The most common treatment for HER2-positive breast cancer is chemotherapy plus HER2-directed therapy. This is followed by surgery, then continues with HER2-directed therapy. This is often best for patients with large tumors or cancer in regional lymph nodes. For targeted therapy, your doctor may recommend a family of drugs commonly known as monoclonal antibodies. This includes trastuzumab (Herceptin), the first-in-its-class precision therapy drug approved by the FDA for HER2+ breast cancer. It's less common, but you may have surgery first, followed by chemotherapy and HER2-directed therapy. Your doctor may choose this sequence if you have a small tumor that isn't in your lymph nodes. Your doctor may also recommend endocrine therapy. This treatment involves taking a daily pill for at least 5 years after you complete chemotherapy and surgery.	[('HER2+ metastatic breast cancer', 'has recommended treatment of', 'chemotherapy plus HER2-directed therapy'), ('HER2-directed therapy', 'targets', 'HER2 protein'), ('HER2 protein', 'located on', 'cancer cells'), ('chemotherapy plus HER2-directed therapy', 'is', 'first line of treatment')]	[('HER2-positive breast cancer', 'is described as', 'aggressive cancer'), ('HER2-positive breast cancer', 'has', 'advances in treatment'), ('HER2-positive breast cancer', 'requires', 'unique treatment approach'), ('HER2-positive breast cancer', 'is different from', 'other breast cancer types'), ('Treatment', 'is influenced by', 'tumor size'), ('Treatment', 'is influenced by', 'cancer metastasis status'), ('Treatment', 'is influenced by', 'risk of recurrence'), ('Common treatment', 'consists of', 'chemotherapy plus HER2-directed therapy'), ('Treatment sequence', 'follows', 'surgery'), ('Common treatment', 'is best for', 'patients with large tumors'), ('Common treatment', 'is best for', 'cancer in regional lymph nodes'), ('Monoclonal antibodies', 'includes', 'trastuzumab (Herceptin)'), ('Trastuzumab', 'is', 'first-in-its-class precision therapy drug'), ('Trastuzumab', 'is approved by', 'FDA'), ('Trastuzumab', 'is approved for', 'HER2+ breast cancer'), ('Alternative treatment sequence', 'starts with', 'surgery'), ('Alternative treatment sequence', 'is followed by', 'chemotherapy and HER2-directed therapy'), ('Alternative treatment sequence', 'is suitable for', 'small tumors without lymph node involvement'), ('Endocrine therapy', 'involves', 'daily pill'), ('Endocrine therapy', 'duration', 'at least 5 years'), ('Endocrine therapy', 'starts after', 'completion of chemotherapy and surgery')]	0
Bariatric surgery leads to positive outcomes in mental health.	IMPORTANCE Bariatric surgery is associated with sustained weight loss and improved physical health status for severely obese individuals. Mental health conditions may be common among patients seeking bariatric surgery; however, the prevalence of these conditions and whether they are associated with postoperative outcomes remains unknown. OBJECTIVE To determine the prevalence of mental health conditions among bariatric surgery candidates and recipients, to evaluate the association between preoperative mental health conditions and health outcomes following bariatric surgery, and to evaluate the association between surgery and the clinical course of mental health conditions. DATA SOURCES We searched PubMed, MEDLINE on OVID, and PsycINFO for studies published between January 1988 and November 2015. Study quality was assessed using an adapted tool for risk of bias; quality of evidence was rated based on GRADE (Grading of Recommendations Assessment, Development and Evaluation) criteria. FINDINGS We identified 68 publications meeting inclusion criteria: 59 reporting the prevalence of preoperative mental health conditions (65,363 patients) and 27 reporting associations between preoperative mental health conditions and postoperative outcomes (50,182 patients). Among patients seeking and undergoing bariatric surgery, the most common mental health conditions, based on random-effects estimates of prevalence, were depression (19% [95% CI, 14%-25%]) and binge eating disorder (17% [95% CI, 13%-21%]). There was conflicting evidence regarding the association between preoperative mental health conditions and postoperative weight loss. Neither depression nor binge eating disorder was consistently associated with differences in weight outcomes. Bariatric surgery was, however, consistently associated with postoperative decreases in the prevalence of depression (7 studies; 8%-74% decrease) and the severity of depressive symptoms (6 studies; 40%-70% decrease). CONCLUSIONS AND RELEVANCE Mental health conditions are common among bariatric surgery patients-in particular, depression and binge eating disorder. There is inconsistent evidence regarding the association between preoperative mental health conditions and postoperative weight loss. Moderate-quality evidence supports an association between bariatric surgery and lower rates of depression postoperatively.	[('Bariatric surgery', 'leads to', 'positive outcomes'), ('positive outcomes', 'occurs in', 'mental health'), ('Bariatric surgery', 'improves', 'mental health')]	[('Bariatric surgery', 'associated with', 'sustained weight loss'), ('Bariatric surgery', 'associated with', 'improved physical health status'), ('Bariatric surgery', 'leads to', 'postoperative decreases in depression'), ('Depression', 'has prevalence of', '19%'), ('Binge eating disorder', 'has prevalence of', '17%'), ('PubMed', 'used as', 'data source'), ('MEDLINE on OVID', 'used as', 'data source'), ('PsycINFO', 'used as', 'data source'), ('Mental health conditions', 'common in', 'bariatric surgery patients'), ('Depression', 'is type of', 'mental health condition'), ('Binge eating disorder', 'is type of', 'mental health condition'), ('GRADE criteria', 'used for', 'quality of evidence rating'), ('Mental health conditions', 'has unclear association with', 'postoperative weight loss'), ('Bariatric surgery', 'reduces', 'depression prevalence'), ('Bariatric surgery', 'reduces', 'depressive symptoms')]	1

Figure 8: Samples of benchmark data. Each sample consists of a claim, its corresponding document, and the extracted KGs (claim_kg and doc_kg), along with the assigned label (Support or Unsupport).

Claim	Doc	Claim_kg	Doc_kg	Label
Hunter Biden seeks subpoenas against Trump for alleged case pressure by citing various sources.	Hunter Biden's attorney Abbe Lowell argued the information was essential to his defense that the case is "possibly, a vindictive or selective prosecution arising from an unrelenting pressure campaign beginning in the last administration," that violated his rights. The subpoena request is before U.S. District Judge Maryellen Noreika, a Trump nominee whose questions about a proposed plea deal over the summer ended with the agreement imploding in July. Hunter Biden had been expected to plead guilty to misdemeanor tax charges in an agreement that would have spared him prosecution on a gun count if he stayed out of trouble for two years. It had been pilloried as a "sweetheart deal" by Trump and congressional Republicans investigating nearly every aspect of Hunter Biden's business dealings and the Justice Department's handling of the case. Hunter Biden has taken a more aggressive legal approach in recent months, striking back with lawsuits against Republican Trump allies who have traded and passed around private data from a laptop that purportedly belonged to him. No new tax charges have yet been filed, but the special counsel overseeing the case has indicated they are possible in Washington or in California, where Hunter Biden lives.	[('Hunter Biden', 'seeks', 'subpoenas'), ('Hunter Biden', 'against', 'Trump'), ('Hunter Biden', 'citing', 'various sources'), ('Hunter Biden', 'alleged case pressure by', 'Trump')]	[('Hunter Biden', 'has attorney', 'Abbe Lowell'), ('Abbe Lowell', 'argued', 'importance of information for defense'), ('Abbe Lowell', 'stated', 'case is possibly vindictive or selective prosecution'), ('Abbe Lowell', 'mentioned', 'unrelenting pressure campaign starting in previous administration'), ('Abbe Lowell', 'asserted', 'pressure campaign violated rights'), ('Subpoena request', 'before', 'U.S. District Judge Maryellen Noreika'), ('Maryellen Noreika', 'nominee of', 'Trump'), ('Maryellen Noreika', 'questioned about plea deal proposal', 'summer'), ('Hunter Biden', 'expected to plead guilty to', 'misdemeanor tax charges'), ('Plea deal', 'would have spared prosecution on gun count', 'if stayed out of trouble for two years'), ('Plea deal', 'criticized as', 'sweetheart deal'), ('Sweetheart deal', 'criticized by', 'Trump and congressional Republicans'), ('Hunter Biden', 'filed lawsuits against', 'Republican Trump allies'), ('Private data laptop', 'purportedly belonged to', 'Hunter Biden')]	0
Johnson's spending bill tensions reveal deep divides within Republican ranks and with Democrats.	Johnson's office did not respond to a request for comment. McCarthy was ousted by eight hardliners on Oct. 3, after averting an Oct. 1 shutdown with a stopgap bill that ran afoul of the far right but won overwhelming support from Democrats. Johnson, who has commanded respect within the far right as an outspoken Christian conservative, irked hardliners this week with his own short-term spending bill to maintain existing government funding levels and programs into early 2024. The bill passed the House with support from 209 Democrats but only 127 Republicans - a troubling sign for the new speaker. He had also angered hardliners by suspending House rules to circumvent their hopes of blocking debate on the measure.	[("Johnson's spending bill", 'reveals', 'deep divides'), ('deep divides', 'within', 'Republican ranks'), ('deep divides', 'with', 'Democrats')]	[('Johnson', 'did not respond to', 'request for comment'), ('McCarthy', 'was ousted by', 'eight hardliners'), ('eight hardliners', 'ousted', 'McCarthy'), ('McCarthy', 'ousted on', 'Oct. 3'), ('McCarthy', 'averted an Oct. 1 shutdown', 'with a stopgap bill'), ('stopgap bill', 'ran afoul of', 'the far right'), ('stopgap bill', 'won overwhelming support from', 'Democrats'), ('Johnson', 'commanded respect within the far right', 'as an outspoken Christian conservative'), ('Johnson', 'irked hardliners this week with', 'his own short-term spending bill'), ('short-term spending bill', 'passed the House with support from', '209 Democrats'), ('short-term spending bill', 'only 127 Republicans supported', ''), ('short-term spending bill', 'troubling sign for', 'new speaker'), ('Johnson', 'angered hardliners by', 'suspending House rules'), ('House rules', 'circumvented', "hardliners' hopes of blocking debate on the measure")]	0
He composed the music to the national anthem of Greenland.	A renowned composer's creation has found a lasting place among the patriotic symbols of Greenland, a country nestled between the frosty expanses of the Arctic and the vast domains of the Atlantic oceans. This melody, adopted with reverence, has since become the backbone of Greenland's official national anthem, a tune steeped in the nation's rich and storied fabric. Cultural representatives in Greenland recently came together to mark the significant anniversary of their national anthem, which, when translated into English, bears the profound title "Our Country, Who's Become So Old." In a ceremonial gathering steeped in tradition and pride, officials took a moment to underscore the importance of continuity by highlighting that the anthem's indigenous name, a deep-seated emblem of Greenlandic identity, has been preserved in its original form since the very day of its inception.	[('Greenland', 'has', 'national anthem'), ('He', 'composed the music to', "Greenland's national anthem")]	[('Greenland', 'has national anthem', 'Melody'), ('Melody', 'adopted with reverence', ''), ('Melody', 'became', "backbone of Greenland's official national anthem"), ("Greenland's national anthem", 'translated into English', "Our Country, Who's Become So Old"), ('Cultural representatives', 'marked the significant anniversary of', "Greenland's national anthem"), ("Greenland's national anthem", 'has indigenous name', ''), ('Indigenous name', 'preserved in its original form', 'since the very day of its inception'), ('Indigenous name', 'deep-seated emblem of', 'Greenlandic identity')]	1
Born on June 1, 1929, in East London, Neville Price was a South African long jumper who competed in the 1952 Summer Olympics.	In the summer of 1929, a warm celebration greeted the Price family as they welcomed a baby boy into their home within the vibrant, industrial folds of East London, on the city's bustling eastern frontier. Birth records chronicling the era's new arrivals pinpoint that, on the outset of June, Neville stood alone as the district's only registered male infant. The threads of history weave forward to reveal, according to extant athletic archives, a sportsman by the name of Neville Price partaking in the fervor of international competition in the summer of 1952. Concurrently, the Olympic Games captivated the global sports audience, standing unchallenged as the season's premier athletic showdown. Further corroboration arrives via the roster of the Melbourne Games, which enumerates a Neville Price among its entrants, contending specifically within the demanding track and field arena. Neville's prowess on the field, especially in the disciplines involving leaps and bounds, had garnered commendation, spotlighting a South African athlete whose flair for track and field events, particularly jumping, was noted as exceptional. Just last year, the athletics community convened in celebration of Neville Price's legacy, marking the anniversary of a national record he set in the long jump category, an accolade that cements his athletic contributions in the annals of sports history.	[('Greenland', 'has', 'national anthem'), ('He', 'composed the music to', "Greenland's national anthem")]	[('Neville Price', 'welcomed into', 'Price family'), ('Price family', 'resided in', 'East London'), ('East London', 'located on', 'city's bustling eastern frontier'), ('Neville Price', 'born on', 'June 1st'), ('Neville Price', 'participated in', 'international competition in the summer of 1952'), ('Neville Price', 'competed in', 'track and field arena'), ('Neville Price', 'excelled in', 'disciplines involving leaps and bounds'), ('Neville Price', 'was recognized for', 'flair for track and field events, particularly jumping'), ('Neville Price', 'set a national record in', 'long jump category'), ('Neville Price', 'celebrated for his legacy in', 'athletics community'), ('Neville Price', 'contributed to', 'sports history through his achievements')]	1

Figure 9: Samples of training data.

Extract a knowledge graph (KG) from the following text. Follow these steps:

1. **Entities**: Identify all entities in the text. Ensure each entity is precise and specific.
2. **Relations**: Extract relationships between entities as triples: ["entity1", "relation", "entity2"].
3. **Coreference Resolution**: Unify references to the same entity (e.g., "Apple Inc." and "Apple" should be the same entity).

Important Requirements:
- The KG must not be empty. Ensure at least one triple is extracted.
- All entities mentioned in the text must be included in the KG, either as part of a triple or as a standalone entity if no relation is found.
- If no explicit relation is found between entities, create a generic relation like "related to" or "associated with" to ensure all entities are connected.
- Each triple must have three non-empty elements: ["entity1", "relation", "entity2"]. None of these elements can be empty or null.

Please only return the KG as a Python list of triples. For example:

<python>
[
["Apple Inc.", "founded by", "Steve Jobs"],
["Apple Inc.", "headquartered in", "Cupertino, California"],
["Apple Inc.", "produces", "iPhone"],
["Steve Jobs", "associated with", "Cupertino, California"]
]
</python>

Text: {**TEXT**}

TEXT
New AI model detects asymptomatic Covid-19 infections through device-recorded coughs Vocal sentiments Prior to the pandemics onset, research groups already had been training algorithms on cellphone recordings of coughs to accurately diagnose conditions such as pneumonia and asthma. The model identified 98.5 percent of coughs from people confirmed with Covid-19, and of those, it accurately detected all of the asymptomatic coughs. When they fed the model new cough recordings, it accurately identified 98.5 percent of coughs from people who were confirmed to have Covid-19, including 100 percent of coughs from asymptomatics who reported they did not have symptoms but had tested positive for the virus.

Extracted Triples
<python>
[
('AI model', 'detects', 'asymptomatic Covid-19 infections'),
('AI model', 'uses', 'device-recorded coughs'),
('research groups', 'trained', 'algorithms'),
('algorithms', 'diagnose', 'pneumonia'),
('algorithms', 'diagnose', 'asthma'),
('algorithms', 'analyze', 'cellphone recordings of coughs'),
('AI model', 'identified', 'Covid-19 coughs'),
('AI model', 'detection accuracy', '98.5 percent'),
('AI model', 'detection accuracy for asymptomatics', '100 percent'),
('asymptomatic patients', 'tested positive for', 'Covid-19'),
('asymptomatic patients', 'reported', 'no symptoms')
]
</python>

Figure 10: Prompt for Triplets Extraction.

Question: Does the Document support the Claim? Please Answer in one word in the form of \'support\' or \'unsupport\'.

Document:
{**DOCUMENT_TEXT**}

Claim:
{**CLAIM_TEXT**}

Question: Does the Document support the Claim? Please Answer in one word in the form of \'support\' or \'unsupport\'.

Document:
{**DOCUMENT_TEXT**}

Claim:
{**CLAIM_TEXT**}

Conclusion:
- Return "1" if the given document fully support the claim.
- Return "0" the given document don't support the claim.
- Giving the Final result in the format: "Final Result: 1" for supported or "Final Result: 0" for unsupported.

DOCUMENT TEXT
New AI model detects asymptomatic Covid-19 infections through device-recorded coughs Vocal sentiments Prior to the pandemics onset, research groups already had been training algorithms on cellphone recordings of coughs to accurately diagnose conditions such as pneumonia and asthma. The model identified 98.5 percent of coughs from people confirmed with Covid-19, and of those, it accurately detected all of the asymptomatic coughs. When they fed the model new cough recordings, it accurately identified 98.5 percent of coughs from people who were confirmed to have Covid-19, including 100 percent of coughs from asymptomatics who reported they did not have symptoms but had tested positive for the virus.

CLAIM TEXT
Artificial intelligence model detects asymptomatic covid-19 infections through cellphone-recorded coughs.

OUTPUT
[Final Result: 1]

Figure 11: Prompt for Fact-checking.