Evidence Grounding vs. Memorization: Why Neural Semantics Matter for Knowledge Graph Fact Verification

Ankit Kumar Upadhyay **John S. Erickson** **Deborah L. McGuinness**
Rensselaer Polytechnic Institute
{upadha2,erickj4,dlm}@rpi.edu

Abstract

Knowledge graphs like DBpedia (Lehmann et al., 2015) enable structured fact verification, but the relative contributions of symbolic structure, neural semantics, and evidence grounding remain unclear. We present a systematic study on FACTKG (Kim et al., 2023) (108,675 claims) comparing symbolic, neural, and LLM-based approaches. Our symbolic baseline using 29 hand-crafted features covering graph structure, entity coverage, and semantic relation type achieves 66.54% accuracy, while BERT over linearized subgraphs reaches 92.68% and graph neural networks plateau at 70%, demonstrating that token-level semantics outperform both symbolic features and message passing. Using GPT-4.1-mini to filter training data, budget-matched controls show that token-budget control recovers most of the gap over truncation-dominated inputs, while LLM semantic selection adds +1.31 points beyond lexical heuristics (78.85% filtered vs. 77.54% heuristic vs. 52.70% unfiltered), showing that semantic relevance, not just evidence quantity, governs learnability. Finally, comparing 300 test claims under memorization (claim-only) versus KG-grounded reasoning with chain-of-thought, we find KG grounding improves GPT-4o-mini and GPT-4.1-mini accuracy by 12.67 and 9.33 points respectively, with models citing specific triples for interpretability. These results demonstrate that neural semantic representations and explicit KG evidence grounding are highly effective for robust, interpretable fact verification.

1 Introduction

Knowledge graphs (KGs) such as DBpedia and Wikidata encode entities and relations that are often extracted from large text corpora. They are widely used in search, recommendation, and question answering, and have recently been proposed as a basis for fact verification: given a natural language claim, the system must decide whether the claim is supported or refuted by the KG.

FACTKG (Kim et al., 2023) formalizes this setting by constructing 108,675 claims over DBpedia, each paired with entities and a one-hop subgraph around those entities. Claims are annotated with multiple reasoning types, including single-hop, multi-hop, multi-claim, existence, substitution, and negation. The original FACTKG paper introduces both claim-only baselines and a GEAR-inspired model that retrieves and aggregates subgraphs. Opsahl (Opsahl, 2024) revisits FACTKG and shows that a BERT-base model over linearized single-step subgraphs can achieve 93.49% accuracy, substantially outperforming QA-GNN-style graph neural networks. The same work also reports that ChatGPT-4o, evaluated in a claim-only setting without KG evidence, reaches 76.33% accuracy, suggesting that LLMs are strong memorization-based baselines but leave their use of explicit KG evidence largely unexplored.

Despite this progress, several key questions remain. First, how well can purely feature-based, non-neural reasoning perform on FACTKG? Second, why do graph neural networks underperform text encoders, even though they operate directly on KG structure? Third, can LLMs be used not only as black-box classifiers, but also as semantic filters that improve the quality of KG-based training data by selecting the most relevant triples for each claim? Finally, when we give LLMs explicit KG evidence, how much do they actually gain over claim-only memorization, and to what extent can we inspect their reasoning through evidence attribution?

In this paper we address these questions through a semantics-focused empirical study of FACTKG that compares symbolic, neural, and LLM-based approaches under a shared experimental pipeline.

Contributions. Our work makes the following contributions:

Proceedings of the Ninth Fact Extraction and VERification Workshop (FEVER), pages 28–46
March 28, 2026 ©2026 Association for Computational Linguistics

- **Symbolic baseline.** We construct feature-based symbolic baselines for FACTKG using 29 hand-crafted features spanning graph structure, entity coverage, and relation semantics (detailed in Appendix B). A simple XGBoost model achieves 66.54% test accuracy, establishing a strong symbolic baseline and revealing where interpretable, non-neural methods fail, especially on negation and multi-hop reasoning.
- **Neural encoders vs. GNNs.** We faithfully reproduce the BERT-base baseline of Opsahl (Opsahl, 2024) on linearized one-hop subgraphs (92.68% accuracy) and evaluate QA-GNN and a cross-attention variant, both of which remain around 70%. This confirms that token-level neural semantics over linearized KG evidence currently outperform graph-native message passing for FACTKG.
- **LLM-assisted semantic filtering.** We use GPT-4.1-mini to select the top-k=10 most relevant triples per claim, avoiding truncation from dense one-hop subgraphs. Fine-tuning BERT on 9,645 filtered examples reaches 78.85% accuracy, while the same examples without selection (512-token truncation) achieve 52.70%. Under a fixed k=10 evidence budget, lexical selection recovers most of the truncation gap and GPT-4.1-mini reranking adds a further +1.31 points beyond the heuristic baseline, showing that semantic evidence prioritization yields a consistent additional gain.
- **Memorization vs. KG-grounded LLM reasoning.** We design a 300-example stratified test set balanced across reasoning types and compare GPT-4o-mini and GPT-4.1-mini in two modes: memorization (claims only) and KG-grounded reasoning (claims plus full one-hop subgraphs with chain-of-thought and triple citations). KG grounding improves accuracy from 71.67% to 84.33% for GPT-4o-mini and from 74.67% to 84.00% for GPT-4.1-mini, demonstrating that explicit KG evidence and evidence-aware prompting substantially enhance LLM fact verification.

Overall, our results argue that semantics, both in the sense of KG structure and neural representations, are central to fact verification, and that LLMs are most effective when used as evidence-grounded reasoners and semantic curators, rather than purely as memorization engines.

2 Background and Related Work

2.1 From FEVER to knowledge graph fact verification

Fact verification has emerged as a central task in NLP, driven by the need to combat misinformation at scale. The FEVER dataset (Thorne et al., 2018) established the standard formulation: given a claim and a corpus of evidence, classify the claim as supported, refuted, or not enough information. FEVER's release prompted influential architectures including GEAR (Zhou et al., 2019), which aggregates evidence using graph-based reasoning over sentence-level representations; NSMN (Nie et al., 2019), which employs homogenous neural networks for document retrieval, sentence selection, and claim verification; and GERE (Chen et al., 2022), which unifies retrieval and verification in a single generative step.

Despite its impact, FEVER has documented limitations. Claims exhibit linguistic biases that allow models to achieve strong performance without consulting evidence (Schuster et al., 2019), and models trained on FEVER suffer performance drops when evidence is adversarially modified (Hidey et al., 2020). While augmented test sets (Schuster et al., 2019, 2021) partially address these issues, these findings motivate exploration of alternative evidence formats and datasets.

Structured evidence offers a complementary path. TabFact (Chen et al., 2019) introduced table-based fact verification, demonstrating that models must reason over structured data formats beyond unstructured text. Knowledge graphs (KGs) extend this structured setting by representing entities and relations as interconnected triples. FACTKG (Kim et al., 2023) formalizes KG-based fact verification using DBpedia (Lehmann et al., 2015), constructing 108,675 claims annotated with reasoning types (single-hop, multi-hop, multi-claim, existence, negation). The benchmark model adapts GEAR to KG evidence, using two fine-tuned language models to predict relevant edges and subgraph depth, achieving 77.65% test accuracy. Opsahl (Opsahl, 2024) revisits FACTKG and shows that BERT-base over linearized single-step subgraphs achieves 93.49% accuracy, substantially outperforming QA-GNN and other graph neural networks, while ChatGPT-4o reaches 76.33% in claim-only mode without KG evidence.

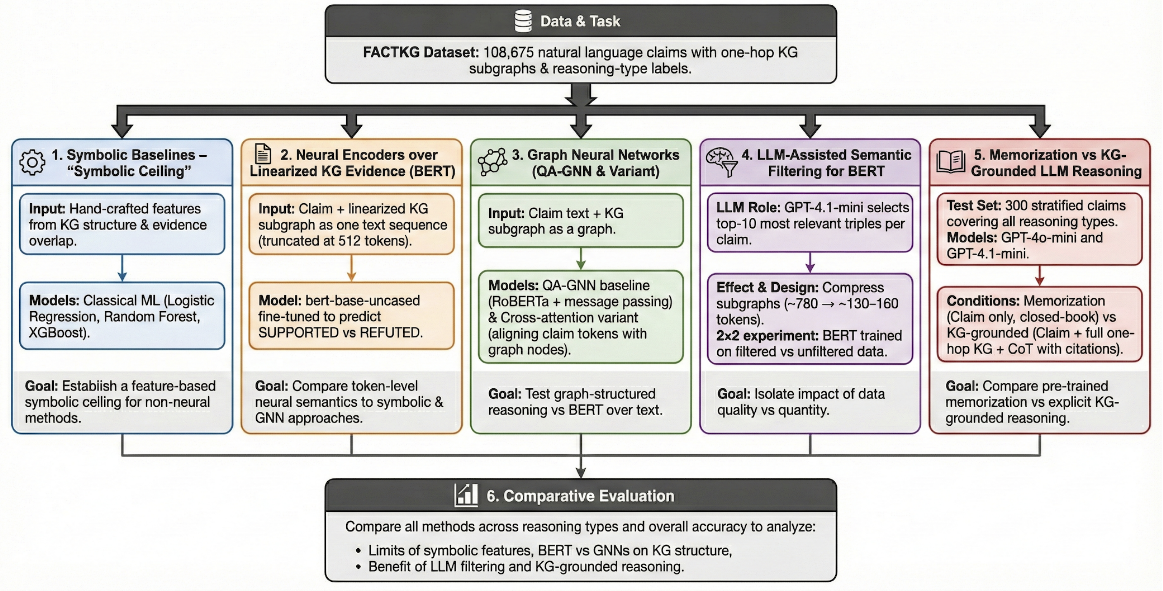

Data & Task
FACTKG Dataset: 108,675 natural language claims with one-hop KG subgraphs & reasoning-type labels.
1. Symbolic Baselines – "Symbolic Ceiling"
Input: Hand-crafted features from KG structure & evidence overlap.
Models: Classical ML (Logistic Regression, Random Forest, XGBoost).
Goal: Establish a feature-based symbolic ceiling for non-neural methods.
2. Neural Encoders over Linearized KG Evidence (BERT)
Input: Claim + linearized KG subgraph as one text sequence (truncated at 512 tokens).
Model: bert-base-uncased fine-tuned to predict SUPPORTED vs REFUTED.
Goal: Compare token-level neural semantics to symbolic & GNN approaches.
3. Graph Neural Networks (QA-GNN & Variant)
Input: Claim text + KG subgraph as a graph.
Models: QA-GNN baseline (RoBERTa + message passing) & Cross-attention variant (aligning claim tokens with graph nodes).
Goal: Test graph-structured reasoning vs BERT over text.
4. LLM-Assisted Semantic Filtering for BERT
LLM Role: GPT-4.1-mini selects top-10 most relevant triples per claim.
Effect & Design: Compress subgraphs (~780 → ~130–160 tokens).
2x2 experiment: BERT trained on filtered vs unfiltered data.
Goal: Isolate impact of data quality vs quantity.
5. Memorization vs KG-Grounded LLM Reasoning
Test Set: 300 stratified claims covering all reasoning types.
Models: GPT-4o-mini and GPT-4.1-mini.
Conditions: Memorization (Claim only, closed-book) vs KG-grounded (Claim + full one-hop KG + CoT with citations).
Goal: Compare pre-trained memorization vs explicit KG-grounded reasoning.
6. Comparative Evaluation
Compare all methods across reasoning types and overall accuracy to analyze:
• Limits of symbolic features, BERT vs GNNs on KG structure,
• Benefit of LLM filtering and KG-grounded reasoning.

Figure 1: Overview of our experimental pipeline comparing symbolic, BERT, GNN, and LLM approaches.

2.2 Hybrid symbolic-neural approaches

Hybrid approaches that combine symbolic knowledge with neural models have been explored in several domains (Mao et al., 2019; Yi et al., 2018). In text-based fact verification, Yuan and Vlachos (Yuan and Vlachos, 2024) use semantic triples extracted from text and external KGs to support zero-shot FEVER-style verification. Their setting addresses sparse evidence by augmenting text with triples. In contrast, FACTKG offers dense KG evidence: one-hop subgraphs can contain dozens of triples per claim, and the challenge becomes selecting the most relevant information rather than augmenting with more.

Our symbolic baseline and LLM-based filtering speak directly to this dense-evidence regime. Symbolic features provide interpretability and a baseline. LLM filtering uses neural semantics to distill large subgraphs into compact, high-signal views that are easier for BERT to learn from. Other recent work includes rule-based systems tailored to FACTKG's reasoning types (Momii et al., 2023) and EVOCA (De Felice et al., 2025), an explainable graph-alignment method.

2.3 LLMs for fact verification and attribution

LLMs have shown strong performance on many reasoning and fact-checking benchmarks (Pan et al., 2023; Wei et al., 2022), but their behavior often reflects a blend of memorization and heuristic reasoning rather than explicit evidence use. Rashkin et al. (Rashkin et al., 2023) argue that trustworthy fact-checking requires attribution: models should ground their claims in identifiable sources.

In FACTKG, this is particularly relevant. Opsahl's ChatGPT-4o experiment (Opsahl, 2024) tests memorization: the model sees claims only and must decide true/false based on its pre-trained parameters. Our KG-grounded experiments instead give the model explicit subgraphs and ask it to reason using chain-of-thought while citing triple indices. This makes the source of each decision inspectable and moves the setup closer to attribution-focused fact verification.

Recent work has begun combining LLMs with KG evidence for fact-checking and question answering. Salnikov et al. (Salnikov et al., 2023) augment LLMs with KG subgraphs for factoid QA, demonstrating that structured evidence improves factual accuracy. Pan et al. (Pan et al., 2023) use program-guided reasoning over KGs for complex fact-checking, showing that explicit reasoning traces improve both accuracy and interpretability. Our contribution differs in that we provide a *controlled comparison* of memorization versus KG-grounded chain-of-thought on the same FACTKG test claims, isolating the effect of explicit evidence provision rather than simply showing that "LLM + KG" yields better numbers. Additionally, our LLM filtering experiments demonstrate that semantic quality of KG evidence matters more than raw quantity for downstream model training.

Table 1: Distribution of reasoning types in the FACTKG test set (9,041 examples). Claims can have multiple type annotations, so percentages sum to more than 100%.

Reasoning Type	Count	Percentage
Single-hop	6,537	72.3%
Multi-claim	3,293	36.4%
Multi-hop	2,073	22.9%
Substitution	856	9.5%
Existence	1,299	14.4%
Negation	1,022	11.3%

3 The FACTKG Dataset

FACTKG (Kim et al., 2023) contains 108,675 claims derived from DBpedia. Each claim is associated with one or more entities and a KG subgraph representing the one-hop neighborhood of those entities. The official splits allocate 86,367 examples for training, 13,267 for validation, and 9,041 for testing.

3.1 Reasoning type taxonomy

Following Opsahl (Opsahl, 2024), we evaluate on six reasoning types: **Existence** (whether a property holds), **Substitution** (entity/value replacement), **Multi-hop** (chaining multiple triples), **Multi-claim** (conjunction of assertions), **Negation** (relation does not hold), and **Single-hop** (any claim not requiring multi-hop reasoning). Claims can have multiple type annotations, and each contributes to accuracy for every type it is tagged with. Table 1 shows the distribution; percentages sum to more than 100% due to this multi-label structure.

Subgraphs in the single-step setting are large: on the test set they contain an average of over 50 triples, and linearization typically yields several hundred tokens. For BERT, this means that many examples exceed the 512-token limit and must be truncated, which motivates our later use of LLM-based filtering.

3.2 Subgraph retrieval for experiments

For all experiments in this work, we use single-step subgraph retrieval following Opsahl (Opsahl, 2024). Single-step retrieval includes every knowledge triple that can be traversed in one step from an entity node mentioned in the claim. Opsahl (Opsahl, 2024) showed this simple approach achieves 93.49% accuracy with BERT, substantially outperforming both alternative retrieval strategies and prior trained baselines. Single-step subgraphs are used consistently across all our approaches: symbolic baselines extract structural features, BERT encodes them as linearized text, GNN models process them as graphs, and LLM experiments reason over them.

Table 2: Symbolic baseline results compared to BERT. Top: Overall performance. Bottom: Per-reasoning-type breakdown showing where symbolic features fail (XGBoost selected as best symbolic model).

Overall Performance				
Model	**Acc.**	**P**	**R**	**F1**
Logistic Reg.	64.73	0.65	0.65	0.65
Random Forest	66.26	0.66	0.66	0.66
XGBoost	**66.54**	**0.67**	**0.66**	**0.66**
BERT (prior)	93.49	0.94	0.91	0.92
BERT (ours)	92.68	0.94	0.91	0.92

Per-Reasoning-Type Accuracy		
Type	**XGBoost**	**BERT**
Existence	63.05	98.15
Substitution	76.85	93.08
Multi-hop	67.39	80.08
Multi-claim	69.88	96.42
Negation	41.10	91.70
Single-hop	76.18	96.43

4 Symbolic Baseline

To quantify what is achievable without neural models, we constructed feature-based symbolic baselines using 29 hand-crafted features extracted from KG structure. These features are dataset-agnostic and could be applied to other KG benchmarks with adaptation. We train classical machine learning models (logistic regression, random forest, XGBoost) on features spanning: (1) *graph structure* (degree statistics, connectivity, components); (2) *entity coverage* (claim entity presence, centrality); and (3) *semantic relation types* (temporal, location, biographical). Complete feature definitions and extraction methodology are in Appendix B.

Table 2 summarizes the results and compares symbolic to neural approaches. **XGBoost** achieves the highest accuracy at 66.54%, establishing a feature-based baseline: it significantly outperforms random guessing but is 26.14 points lower than BERT. Symbolic features are surprisingly competitive on substitution, where simple graph patterns suffice, but they fail badly on negation and multi-hop reasoning.

Feature importance analysis reveals which symbolic patterns correlate with support and refutation. Higher average degree, more edges, and greater relation diversity are associated with supported claims. However, high degree variance, many connected components, and a large number of location relations correlate with refuted claims. These patterns provide some intuition but cannot substitute for full semantic understanding. The complete feature analysis is provided in Appendix A. These results show that symbolic features capture certain regularities of FACTKG but cannot model negation, compositional semantics, or fine-grained linguistic nuances. They set a strong but clearly insufficient baseline for fact verification.

5 Neural Encoders over Linearized KG Evidence

We now turn to neural encoders that operate on text. Following prior work (Opsahl, 2024), we treat the claim and its subgraph as a single sequence: the claim text followed by a linearized list of triples. The subgraph is serialized by converting each triple into a short text fragment and concatenating them with separators.

5.1 Model and training

We use `BERT-base-uncased` as the encoder. Inputs are tokenized with a maximum length of 512 tokens; longer sequences are truncated. The [CLS] token's final hidden state is passed to a linear classifier to predict SUPPORTED or REFUTED. We fine-tune all BERT parameters with AdamW, batch size 4, learning rate 5×10^{-6}, and early stopping on validation accuracy. We follow exact hyperparameters and single-step retrieval setting from prior work, using the same train/validation/test splits.

5.2 Results and comparison to symbolic baselines

Our reproduction achieves 92.68% test accuracy, very close to the 93.49% reported in prior work (Opsahl, 2024). Table 3 summarizes performance by reasoning type.

Compared to the symbolic baseline of 66.54%, BERT gains roughly 26 points in accuracy. The largest gains occur in negation and multi-hop reasoning, where symbolic features are weakest. This confirms that token-level neural semantics and attention over the linearized subgraph are critical capabilities for FACTKG.

Table 3: BERT baseline reproduction on FACTKG with single-step subgraphs. The model is strong across all reasoning types, including negation.

Type	Acc.	P	R	F1
Existence	98.15	0.977	0.986	0.981
Substitution	93.08	0.345	0.787	0.480
Multi-hop	80.08	0.860	0.752	0.802
Multi-claim	96.42	0.947	0.969	0.958
Negation	91.70	0.917	0.900	0.909
Single-hop	96.43	0.959	0.966	0.962
Overall	92.68	0.936	0.912	0.924

Note: Precision, recall, and F1 are computed for the SUPPORTED class only. Accuracy reflects overall correctness across both classes.

6 Graph Neural Networks: QA-GNN and Variants

Graph neural networks (GNNs) seem like a natural fit for KGs. QA-GNN (Yasunaga et al., 2021) is a hybrid model that encodes the question (or claim) with a language model and performs message passing over a retrieved subgraph. Prior work (Opsahl, 2024) adapts QA-GNN to FACTKG, but finds that it underperforms BERT.

6.1 Architecture

We follow the QA-GNN setup. The claim is encoded with `BERT-base-uncased`, and each entity node in the subgraph is initialized with a precomputed embedding derived from its label. A graph attention network performs several rounds of message passing over the subgraph to produce updated node embeddings. Following QA-GNN, node representations are aggregated via graph pooling and combined with the claim representation in a final classifier to predict the label; additionally, the graph signal is modulated by a claim-conditioned relevance weighting (cosine similarity between the claim [CLS] embedding and node features) prior to message passing.

We additionally evaluated a *cross-attention fusion* variant as a diagnostic baseline for text–graph interaction. In this model, we remove the cosine-based node reweighting and replace the final concatenation with a learned bidirectional attention fusion module between the pooled graph representation and the claim [CLS] embedding before classification. This introduces attention-based fusion of text and graph information, but it *does not implement explicit token–node alignment* between individual claim tokens and specific graph entities, since attention is applied over aggregated repre-

Table 4: GNN baselines on FACTKG. Even with cross-attention, QA-GNN variants lag behind BERT by more than 20 points.

Model	Acc.	F1	Neg. Acc.
QA-GNN (baseline)	69.64	0.675	50.61
QA-GNN (variant)	69.74	0.698	48.40
BERT (comparison)	92.68	0.924	91.70

sentations rather than token-to-node pairs. Future work should compare against joint text-graph architectures such as GreaseLM (Zhang et al., 2021) and DRAGON (Yasunaga et al., 2022), which are specifically designed to learn token-node alignment during pretraining.

6.2 Results and analysis

Table 4 summarizes the performance of QA-GNN and another variant, along with BERT for comparison.

Both GNN variants are stronger than purely symbolic baselines but far below BERT. Negation accuracy hovers around 50%, indicating that message passing over graph edges is not sufficient to capture the semantics of negation and absence. Multi-hop reasoning also remains challenging.

Why do GNNs underperform? We hypothesize that QA-GNN's underperformance stems from two factors. First, message passing over graph edges does not naturally encode negation or absence of relations; a GNN sees what edges exist but struggles to reason about what is *missing*. Second, the linearized text representation allows BERT's attention mechanism to directly compare claim tokens with evidence tokens at a fine granularity, whereas GNN node aggregation loses this token-level alignment. The particularly poor performance on negation (Table 4) supports this interpretation. These results replicate and extend the core finding: in FACTKG, text-based encoders with neural semantics outperform graph-native architectures.

7 LLM-Assisted Semantic Filtering for Efficient BERT Training

FACTKG single-step subgraphs are dense: when all triples are linearized, many inputs exceed BERT's 512-token limit and must be truncated. This discards potentially useful evidence and makes training more expensive. We therefore explore using an LLM as a semantic filter that selects only the most relevant triples per claim before encoding them with BERT.

Table 5: Token and truncation statistics for FACTKG under single-step subgraph linearization. Subgraph length refers to the linearized KG evidence only (excluding the claim). Truncation is measured after concatenating claim, [SEP], and subgraph with a maximum length of 512 tokens.

Split	#Ex	Mean	Med	Tri	Trunc
Train	86,367	760.5	489.0	57.8	49.9%
Dev	13,267	652.3	435.0	50.0	45.4%
Test	9,041	687.3	504.0	52.5	51.0%

7.1 Motivation and token statistics

To understand how severe truncation is, we ran a dedicated analysis script over the official FACTKG splits using the BERT tokenizer. For each example, the script tokenizes the claim, the linearized subgraph, and their concatenation, and then records sequence lengths, triple counts, and whether truncation would occur at 512 tokens.

Table 5 summarizes the main statistics. On all three splits, mean subgraph lengths are well above 600 tokens, and roughly half of all examples exceed the 512-token limit and are truncated. The script estimates that each triple contributes approximately 13 tokens on average.

Using this estimate, we can predict the approximate total length if we kept only the top-k triples per subgraph. The choice of $k = 10$ emerges as a natural compromise: it reduces average input length by roughly 77–80% across splits while keeping enough context for most claims. This ensures that no filtered example exceeds 512 tokens.

7.2 Method

We use GPT-4.1-mini as a semantic filter over the FACTKG subgraphs. For each claim, we provide the claim text and the full list of its KG triples, and ask the model to select the ten most relevant triples for fact verification, returning only their indices. Subgraphs are linearized using a helper function that strips URI prefixes, replaces underscores with spaces, and formats each triple.

Filtering is applied to a stratified subset of the training data. Starting from the 86,367 training examples, we sample 9,645 claims with approximately balanced coverage of the main reasoning types. For each sampled claim, GPT-4.1-mini produces a ranked list of relevant triples; we retain the top ten. In the filtered subset, the average number of triples per subgraph drops from 57.8 to about 9, and average token length falls from roughly 780 to approximately 130–160 tokens. After filtering, no example exceeds BERT's 512-token limit.

For comparison, we also construct an unfiltered subset of the same 9,645 claims with full subgraphs. The same filtering procedure is applied to the test data to obtain a "clean" evaluation split. Complete implementation details, including the three-stage filtering pipeline (prefiltering, LLM scoring, rank fusion), the exact LLM prompt, entity-set alignment methodology, and stratified sampling procedure, are provided in Appendix D.

7.3 Experimental design

We train four BERT models in a controlled 2×2 design crossing training data type (filtered vs. unfiltered) with test data type (original vs. filtered), using the same 9,645 claims in all cases. All models use `bert-base-uncased`, max length 512, batch size 16, learning rate 2×10^{-5}, and early stopping on validation loss.

7.4 Results

Table 6 summarizes overall results and the per-reasoning-type breakdown for Option A models.

The contrast between OptA-Filt. (78.85%) and OptA-Unfilt. (52.70%) is especially striking: both models are trained on the same 9,645 examples, but OptA-Filt. applies explicit evidence selection via method described in Section 7.2 while OptA-Unfilt. uses no selection and is dominated by BERT's 512-token truncation. To isolate what drives this 26.15-point gap, we introduce three OptA evidence controls that hold the evidence budget constant at k=10 triples (Table 6, *OptA Controls*) while varying only the *Selection / Setting*: (i) *Global Random*, which samples 10 triples randomly from the full subgraph (no selection signal); (ii) *Random-K*, which samples 10 triples from a lexically prefiltered pool of 24; and (iii) *Heuristic*, which deterministically selects the top 10 using must-keep rules and lexical overlap (Jaccard), bypassing LLM scoring entirely.

Table 6: Effect of filtering strategies on BERT. **Top:** Overall performance, including original conditions and three k=10 evidence controls for OptA (9,645 training examples) that isolate budget, lexical ordering, and LLM selection effects. **Bottom:** Per-reasoning-type breakdown across OptA variants.

Overall Performance			
Condition	**Selection / Setting**	**Test Set**	**Acc.**
Main Conditions			
OptA-Filt.	**LLM rerank (k=10)**	**Original**	**78.85**
OptA-Unfilt.	No sel. (512 trunc.)	Original	52.70
OptB-Filt.	LLM rerank (k=10)	Filtered	76.00
OptB-Unfilt.	No sel. (512 trunc.)	Filtered	63.27
Full BERT	Unfiltered (Full Data)	Original	92.68
OptA Controls (k=10, same budget)			
Heuristic	Top-k (must-keep+Jacc.)	Original	77.54
Random-K	Local rand (pool-24)	Original	74.97
Global Random	Global rand (full subg.)	Original	73.48

Per-Type Accuracy (OptA Variants)

Type	**Unfilt.**	**Global**	**Rand-K**	**Heur.**	**OptA-Filt.**
Existence	56.58	88.45	88.68	89.22	**91.15**
Substitution	**95.03**	84.90	84.93	73.12	81.69
Multi-hop	47.71	74.29	74.92	**76.41**	75.45
Multi-claim	57.79	71.94	74.52	78.32	**80.66**
Negation	61.11	73.82	73.90	76.03	**76.18**
Single-hop	54.19	73.23	74.99	77.87	**79.87**
Overall	52.70	73.48	74.97	77.54	**78.85**

Under this fixed k=10 budget, the gap decomposes as follows. Moving from truncation-dominated inputs to a fixed-size budget yields the largest jump (OptA-Unfilt. → Global Random: 52.70 → 73.48, +20.78). Within the constant-budget tier, lexical selection adds a further +4.06 points (Global Random → Heuristic: 73.48 → 77.54), and LLM reranking contributes an additional +1.31 points beyond the heuristic baseline (Heuristic → OptA-Filt.: 77.54 → 78.85). Thus, while the full 26.15-point improvement reflects both truncation effects and selection, the k=10 controls isolate a smaller but consistent gain from LLM-based semantic selection beyond lexical ordering.

Per-type results in Table 6 reinforce two conclusions. First, controlling truncation is necessary but not sufficient: moving from OptA-Unfilt. (512-token truncation with no selection) to a fixed k=10 budget yields large gains for evidence-hungry types such as Existence (56.58→88.45) and Multi-hop (47.71→74.29), indicating that the unfiltered setting primarily fails due to signal loss under truncation. Second, within the constant-budget tier, *LLM semantic filtering provides the strongest evidence prioritization and achieves the best overall and per-type performance in most cases*. OptA-Filt. attains the top accuracy on Existence (91.15), Multi-claim (80.66), Negation (76.18), and Single-hop (79.87), and remains competitive on Multi-hop (75.45 vs. 76.41 for heuristic). Substitution is the only exception where OptA-Unfilt. is highest (95.03), consistent with entity-swap claims depending on early-walk triples that survive truncation. Overall, these trends suggest that *semantic filtering improves signal-to-noise*: lexical heuristics recover most of the benefit once the budget is fixed, while LLM scoring provides a consistent additional boost on the harder reasoning categories where evidence choice matters most.

8 LLMs for Memorization vs. KG-Grounded Reasoning

We now address the role of LLMs in FACTKG. Prior work (Opsahl, 2024) evaluates ChatGPT-4o on claims without KG evidence and reports 76.33% accuracy. This tests the model's pre-trained knowledge, not its ability to reason over the KG itself. Our goal is to compare this memorization mode with a KG-grounded mode where the model is given the claim, its one-hop subgraph, and few-shot chain-of-thought examples.

8.1 Experimental setup

Gold few-shot examples. We use 10 gold-standard few-shot examples covering all reasoning types. Each contains a claim, its full unfiltered KG subgraph, a verdict (SUPPORTED/REFUTED), an explanation citing triple indices, and key evidence indices.

Test set construction. We perform stratified sampling over reasoning types from the 9,041 test claims to obtain 300 examples with random seed 42. We keep only examples with at least 10 triples. The distribution: 70 existence, 132 substitution, 89 multi-hop, 99 multi-claim, 92 negation (overlapping).

Models and prompting. We evaluate GPT-4o-mini and GPT-4.1-mini with temperature 0 and seed 42. For each test example, we construct one of two prompts:

- **Memorization prompt**: Claim only, no KG evidence. Model outputs JSON with verdict (True/False) and explanation based on pre-trained knowledge.
- **KG-grounded prompt**: Claim plus full unfiltered subgraph with chain-of-thought examples. Model outputs JSON with verdict (SUPPORTED/REFUTED) and explanation citing triple IDs.

Full prompt templates are in Appendix C.

Table 7: Memorization vs. KG-grounded reasoning on 300 stratified FACTKG claims (paired). Accuracies include 95% Wilson CIs; Δ is KG−Mem with McNemar exact p-values.

Model	Mem Acc	KG Acc	Δ
	(95% CI)	(95% CI)	(p-value)
GPT-4o-mini	71.67 (66.3–76.5)	84.33 (79.8–88.0)	+12.67 6.3×10^{-5}
GPT-4.1-mini	74.67 (69.5–79.3)	84.00 (79.4–87.7)	+9.33 1.1×10^{-3}

8.2 Results

Table 7 shows overall accuracy for each model and condition, while Table 8 breaks down performance by reasoning type. Both models benefit substantially from KG grounding: GPT-4o-mini improves from 71.67% to 84.33% (+12.67 points, $p = 6.27\times10^{-5}$) and GPT-4.1-mini from 74.67% to 84.00% (+9.33 points, $p = 1.09\times10^{-3}$), with both improvements statistically significant under paired McNemar exact tests; paired bootstrap 95% confidence intervals for Δ remain positive (GPT-4o-mini: [6.67, 18.67], GPT-4.1-mini: [4.00, 14.67]) and accuracy CIs are reported in Table 7. Full paired counts and uncertainty are reported in Table 13 in Section E.5. Our GPT-4.1-mini memorization baseline (74.67%) is consistent with prior work's ChatGPT-4o claim-only result (76.33%) (Opsahl, 2024), given model differences. Critically, KG grounding pushes both models to 84% on the *same* 300 claims, demonstrating that the main limiting factor in claim-only settings is absence of explicit evidence rather than model capacity. Per-reasoning-type analysis indicates that existence claims contribute the largest gains for both models (both $p < 0.001$), while multi-hop claims show no improvement under KG grounding for GPT-4.1-mini ($\Delta = 0.00$, $p = 1.0$), suggesting that the one-hop subgraph may lack the intermediate links required for multi-step chains; whether this reflects a structural limitation of single-hop retrieval or a model-specific failure to chain available evidence is left to future work. We treat remaining per-type differences as descriptive given smaller n and multiple comparisons.

9 Error Analysis

We analyzed all 300 LLM comparison examples to understand failure modes. KG grounding rescued 49 cases where memorization failed, while memorization rescued only 21 cases where KG

Table 8: Per-type accuracy for GPT-4o-mini (4o-m) and GPT-4.1-mini (4.1-m) on 300 stratified claims. Parentheses show the number of claims tagged with each (multi-label) reasoning type. Significance markers are paired McNemar exact tests: $^{***}p<0.001$, $^{**}p<0.01$, $^{*}p<0.05$.

	GPT-4o-m			GPT-4.1-m		
Reasoning Type (n)	**M**	**KG**	**Δ**	**M**	**KG**	**Δ**
Existence (70)	57.14	91.43	+34.29***	65.71	88.57	+22.86***
Substitution (132)	87.12	80.30	-6.82	87.12	90.15	+3.03
Multi-hop (89)	67.42	77.53	+10.11	73.03	73.03	+0.00
Multi-claim (99)	86.87	84.85	-2.02	82.83	87.88	+5.05
Negation (92)	66.30	76.09	+9.78	70.65	82.61	+11.96
Overall (300)	71.67	84.33	+12.67***	74.67	84.00	+9.33**

grounding failed (2.3:1 ratio), demonstrating net benefits from explicit evidence. KG evidence particularly helps with obscure entities, question-type claims, and formal KG relations, but can mislead on dense multi-hop claims or implicit information. Notably, multi-hop claims show identical 73.03% performance in both modes, suggesting GPT-4.1-mini lacks chaining logic regardless of evidence provision. The test set's label distribution (41% SUPPORTED, 59% REFUTED) reveals bias: KG grounding achieves 93.79% on REFUTED but only 69.92% on SUPPORTED, as detecting absence is easier than confirming presence. All KG-grounded responses include triple citations (mean 6.28 triples cited from 36.7 available), confirming genuine evidence engagement. Detailed per-claim analysis is in Appendix E.

10 Discussion

Across symbolic, neural, and LLM-based methods, a consistent story emerges. Symbolic features plateau at 67% accuracy, failing on semantically complex reasoning types. In our experiments, graph neural networks improve over symbolic baselines but lag BERT by 22+ points, particularly on negation and existence, as message passing cannot capture absence or token-level alignment. BERT-base sets the current state of the art at 92.68%, leveraging token-level processing to model composition, negation, and contextual semantics that graph structure alone cannot provide.

LLMs serve two complementary roles. As *semantic filters*, they dramatically improve training data quality: GPT-4.1-mini filtering enables BERT to learn effectively from 9,645 carefully selected examples, outperforming equally sized unfiltered samples by 26 points. As *KG-grounded reasoners*, LLMs perform fact verification with explicit evidence and chain-of-thought, gaining 9-13 points over memorization while exposing reasoning through triple citations. This comparison reveals that without evidence, models rely on coarse world knowledge and plausibility judgments; with evidence, they anchor decisions in explicit triples, enabling auditable behavior and directly addressing limitations of prior claim-only experiments.

11 Conclusion

This work revisits FACTKG with a focus on semantics. We established a symbolic baseline using hand-crafted features, reproduced and extended BERT and QA-GNN baselines, introduced LLM-based semantic filtering for efficient BERT training, and designed a new 300-example LLM experiment that directly compares memorization to KG-grounded reasoning.

Our findings can be summarized as follows. First, symbolic methods are useful but fundamentally limited on FACTKG, especially for negation and multi-hop reasoning. Second, neural encoders over linearized KG evidence currently offer the best performance among non-LLM architectures. Third, LLMs are most powerful when they are used to improve the semantic quality of data and to reason over explicit KG evidence with chain-of-thought, rather than merely as black-box memorization engines. Overall, our results demonstrate that semantics, both in terms of KG structure and neural representations, are central to fact verification, and that LLMs are most effective as evidence-grounded reasoners and semantic curators, though evidence provision for multi-hop reasoning remains an open challenge.

Future work will include qualitative analysis of the semantically filtered datasets produced by LLMs, deeper analysis of LLM outputs in our KG-grounded prompting setup, joint training of BERT with learned LLM-based filtering, and new architectures that more directly exploit graph structure while preserving the semantic flexibility of transformers; we hope this study encourages further work on semantically grounded, evidence-aware fact verification over knowledge graphs.

12 Limitations

Our work has several limitations. The LLM filtering experiments use only 9,645 training examples (roughly 10% of the full dataset), and our 300-

example LLM comparison represents only 3.3% of the FACTKG test set, which may not capture all edge cases despite stratified sampling. Subsampling was intentionally limited due to computational and financial constraints related to API costs and filtering time. However, our stratified sampling approach ensures the 300-example test set maintains proportional representation of reasoning types, and our 9,645 filtered training examples provide balanced coverage across reasoning categories (see Appendix D), making these subsamples representative despite their size. We evaluate only GPT-4o-mini and GPT-4.1-mini which are not reasoning models; results may differ with other LLM families or reasoning-tuned models. Our interpretability analysis confirms that all KG-grounded responses cite specific triples, but citation presence alone does not validate faithful grounding; manual verification of whether cited triples actually entail or contradict the claim is deferred to future work. Finally, all experiments use one-hop subgraphs; multi-hop claims requiring longer reasoning chains may not have sufficient evidence in these subgraphs, potentially underestimating model capabilities. The multi-hop stagnation for GPT-4.1-mini under KG grounding (Table 8) warrants investigation into whether the one-hop subgraph structure systematically excludes intermediate reasoning links for multi-hop claims. Several directions could address these limitations and extend this work. Replacing BERT with text–KG fusion models like GreaseLM (Zhang et al., 2021) and DRAGON (Yasunaga et al., 2022), which jointly encodes text and graph structure with bidirectional MLM+link prediction, would test whether stronger cross-modal pretraining can close or surpass the current baseline while leveraging dense subgraph structure. Our semantic filtering results motivate models that learn which edges to traverse rather than relying on fixed one-hop expansion, treating evidence selection as a sequential decision problem with graph policy networks rewarded by downstream verification accuracy. Scaling our 300-example LLM evaluation to the full test set, paired with more detailed analysis of model outputs under our KG-grounded prompting setup and experiments with reasoning-tuned models, would clarify the remaining potential for evidence-grounded verification.

13 Ethics Statement

This work uses the publicly available FACTKG dataset derived from DBpedia and does not involve human subjects, personal data, or sensitive data collection. We follow the dataset and knowledge base terms of use. The LLM-assisted components send only dataset text and KG triples to the API and do not include private or user-generated information.

Fact verification systems can help combat misinformation, but they can also be misused for censorship or automated content moderation without appropriate human oversight. Automated verifiers can produce incorrect decisions or overconfident explanations, especially under dataset bias or KG incompleteness. We recommend human review for high-stakes applications.

Our experiments use commercial LLM APIs (GPT-4o-mini, GPT-4.1-mini), which incur computational costs and associated environmental impact. While the scale of our experiments is limited (300 test examples for the main comparison), we acknowledge these concerns would grow with larger evaluations.

References

Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. Gere: Generative evidence retrieval for fact verification. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2184–2189.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification. *arXiv preprint arXiv:1909.02164*.

Lorenzo De Felice, Elena Cabrio, and Serena Villata. 2025. Evoca: Explainable verification of claims by graph alignment. *Information*, 16(1):45.

Christopher Hidey, Tuhin Chakrabarty, Tariq Alhindi, Siddharth Varia, Kriste Krstovski, Mona Diab, and Smaranda Muresan. 2020. Deseption: Dual sequence prediction and adversarial examples for improved fact-checking. *arXiv preprint arXiv:2004.12864*.

Jiho Kim, Sungjin Park, Yeonsu Kwon, Yohan Jo, James Thorne, and Edward Choi. 2023. Factkg: Fact verification via reasoning on knowledge graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 16190–16206.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian

Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*.

Shinya Momii, Naoya Inoue, and Kentaro Inui. 2023. Rule-based fact verification utilizing knowledge graphs. In *Proceedings of the 6th Joint International Conference on Data Science & Management of Data*.

Mark Newman. 2018. *Networks*, 2nd edition. Oxford University Press, Oxford.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6859–6866.

Tobias A Opsahl. 2024. Fact or fiction? Improving fact verification with knowledge graphs through simplified subgraph retrievals. In *Proceedings of the 6th Workshop on Fact Extraction and VERification (FEVER)*, pages 289–298.

Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. 2023. Fact-checking complex claims with program-guided reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 6981–7004.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2023. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840.

Mikhail Salnikov, Le Fang, Aniko Hannak, Hakan Ferhatosmanoglu, and Artem Baklanov. 2023. Large language models meet knowledge graphs to answer factoid questions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin c! robust fact verification with contrastive evidence. *arXiv preprint arXiv:2103.08541*.

Tal Schuster, Darsh J Shah, Yun Jie Serene Yeo, Daniel Filizzola, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. *arXiv preprint arXiv:1908.05267*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 809–819.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D Manning, Percy Liang, and Jure Leskovec. 2022. Deep bidirectional language-knowledge graph pretraining. In *Advances in Neural Information Processing Systems*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 535–546.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B Tenenbaum. 2018. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems*, pages 1039–1050.

Zhaowei Yuan and Andreas Vlachos. 2024. Zero-shot fact verification with semantic triples and knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12844–12865.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2021. Greaselm: Graph reasoning enhanced language models. In *International Conference on Learning Representations*.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. Gear: Graph-based evidence aggregating and reasoning for fact verification. *arXiv preprint arXiv:1908.01843*.

A Feature Importance Details

Table 9 ranks the symbolic features by their importance in our best-performing baseline, XGBoost (66.54% accuracy). Unlike logistic regression, which assigns directional coefficients, XGBoost importance measures the information gain of each feature, effectively quantifying how useful the feature is for distinguishing supported from refuted claims. The dominance of num_connected_components (importance: 0.211) confirms that graph fragmentation is the single most decisive signal for fact verification in this domain.

Table 9: Top 10 most important symbolic features for the XGBoost model (Best Symbolic Baseline).

Feature	Importance	Metric Type
Num. connected components	0.211	Graph Structure
Entities in subgraph	0.097	Entity Matching
Average degree	0.071	Graph Structure
Entities NOT in subgraph	0.065	Entity Matching
Entity coverage	0.059	Entity Matching
Num. claim entities	0.055	Metadata
Inverse relation ratio	0.027	Semantic Type
Has location relations	0.027	Semantic Type
Unique relation types	0.025	Semantic Type
Num. location relations	0.025	Semantic Type

B Symbolic Feature Design and Extraction

This appendix provides complete details on the 29 hand-crafted features used for our symbolic baselines (Section 4). Our feature set combines established graph-theoretic metrics with novel features designed specifically for KG fact verification.

B.1 Feature Categories and Rationale

Our features span four categories, each capturing different aspects of the claim-subgraph relationship:

Graph Structure (9 features). These features capture topological properties of the KG subgraph using standard network analysis metrics (Newman, 2018). We construct a directed graph from the triple list where nodes are entities and edges represent relations. Features include: num_nodes (entity count), num_edges (triple count), graph_density (ratio of existing to possible edges), degree statistics (avg_degree, max_degree, min_degree, degree_std), num_connected_components (graph fragmentation), and has_cycle (presence of circular relation paths).

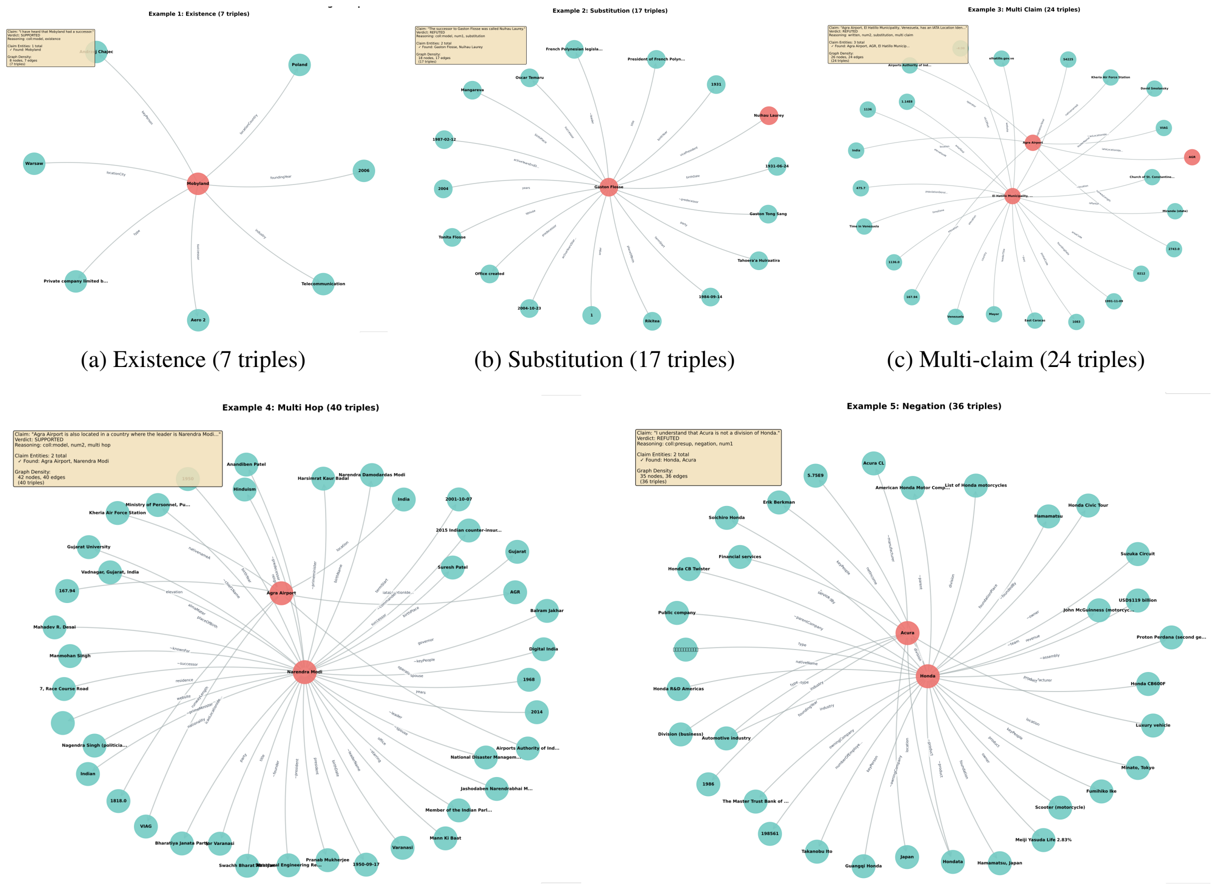
Claim-Entity Matching (7 features). These features measure alignment between the claim and subgraph evidence, inspired by entity linking and question answering systems (Yasunaga et al., 2021). Features include: entity_coverage (fraction of claim entities appearing in subgraph), entities_in_subgraph and entities_not_in_subgraph (absolute counts), edge distribution metrics (avg_edges_per_claim_entity, max_edges_for_claim_entity), and lexical overlap features (claim_words_in_relations, claim_relation_overlap).

Semantic Relation Categories (11 features). These domain-specific features capture whether the subgraph contains relations from predefined semantic categories we designed for DBpedia's schema. We define four categories:

- **Temporal**: birthDate, deathDate, activeYearsStartYear, etc.
- **Location**: location, country, birthPlace, headquarters, etc.
- **Person**: spouse, child, successor, predecessor, etc.
- **Organizational**: team, club, employer, league, etc.

For each category, we extract both binary presence (has_*_relations) and count (num_*_relations) features. Additional semantic features include: relation_type_diversity (Shannon entropy of relation distribution), most_common_relation_freq (dominance of most frequent relation), unique_relation_types (distinct relations), and inverse_relation_ratio (fraction of inverse relations like ~successor).

Metadata (2 features). Basic claim properties: claim_length (word count) and num_claim_entities (entity mentions).


Example 1: Existence (7 triples)
Example 2: Substitution (17 triples)
Example 3: Multi Claim (24 triples)
Example 4: Multi Hop (40 triples)
Example 5: Negation (36 triples)

(a) Existence (7 triples) (b) Substitution (17 triples) (c) Multi-claim (24 triples)

(d) Multi-hop (40 triples) (e) Negation (36 triples)

Figure 2: Example one-hop subgraphs for each reasoning type in FACTKG, showing varying subgraph densities. Claim entities (red/coral nodes) and related entities (teal nodes) with KG evidence.

B.2 Feature Extraction Implementation

Features are extracted from the `walked` column in the FACTKG subgraph data, which contains the list of triples reachable via one-hop traversal from claim entities. For graph structure features, we construct a NetworkX directed graph and compute standard metrics. For semantic features, we perform exact string matching on lowercased relation names (including inverse relations prefixed with ~). For claim-entity matching, we use set operations and Jaccard similarity on (subject, predicate) pairs. Crucially, we rely strictly on input-time subgraph information and do not utilize gold evidence annotations to prevent data leakage.

B.3 Complete Feature List

Table 10 provides the complete list of all 29 features with descriptions and value types.

B.4 Design Rationale

The feature set was designed to answer three questions about symbolic fact verification: (1) Can graph topology alone distinguish supported from refuted claims? (2) How important is entity-claim alignment versus semantic relation types? (3) Are domain-specific relation categories predictive?

Our results (Section 4) show that graph structure and semantic categories are moderately predictive (66.54% accuracy with XGBoost), but fundamental limitations exist: symbolic features cannot capture negation (41.10% accuracy), fine-grained semantics, or compositional reasoning. While additional features could potentially be engineered, the consistent 26-point gap between symbolic baselines and neural methods suggests that semantic understanding requires learned representations rather than hand-crafted features.

Table 10: Complete list of 29 symbolic features for FACTKG.

Feature	**Type**	**Description**
	Metadata (2)	
claim_length	int	Words in claim
num_claim_entities	int	Entities in claim
	Graph Structure (9)	
num_nodes	int	Unique entities
num_edges	int	Triple count
graph_density	float	Edge ratio
avg_degree	float	Mean degree
max_degree	int	Max degree
min_degree	int	Min degree
degree_std	float	Degree std dev
num_connected_components	int	Component count
has_cycle	binary	Has cycles
	Claim-Entity Matching (7)	
entity_coverage	float	Fraction in subgraph
entities_in_subgraph	int	Count present
entities_not_in_subgraph	int	Count missing
avg_edges_per_claim_entity	float	Mean edges
max_edges_for_claim_entity	int	Max edges
claim_words_in_relations	int	Word overlap
claim_relation_overlap	float	Overlap fraction
	Semantic Relations (11)	
has_temporal_relations	binary	Has temporal
num_temporal_relations	int	Temporal count
has_location_relations	binary	Has location
num_location_relations	int	Location count
has_person_relations	binary	Has person
num_person_relations	int	Person count
has_organizational_relations	binary	Has org
num_organizational_relations	int	Org count
relation_type_diversity	float	Shannon entropy
most_common_relation_freq	float	Max frequency
unique_relation_types	int	Distinct types
inverse_relation_ratio	float	Inverse fraction

C LLM Prompt Templates

This appendix lists the exact prompt templates used for the memorization and KG-grounded LLM experiments.

Memorization baseline prompt

```
Task: Determine the truth value (True or False) of the following
claims based on information verifiable from Wikipedia, as represented
in the DBpedia knowledge graph. Provide your answers without using
real-time internet searches or code analysis, relying solely on your
pre-trained knowledge.

Instructions:
- Base your answers solely on your knowledge as of your last training cut-off
- Respond with True for verifiable claims, and False otherwise
- Include a brief explanation for each answer
- If the claim is vague or lacks specific information, please make
  an educated guess

Output Format: JSON with "verdict" and "explanation"

[Few-shot examples omitted for brevity]

Now evaluate this claim based on your pre-trained knowledge:

Claim: {test_claim}

Output JSON (respond with ONLY valid JSON, no other text):
{
  "verdict": "True" or "False",
  "explanation": "Your reasoning based on pre-trained knowledge (2-3 sentences)"
}
```

KG-grounded reasoning prompt

```
You are an expert fact verification system using knowledge graph evidence.

Task: Determine if claims are SUPPORTED or REFUTED based ONLY on the
provided evidence triples.

Instructions:
- Reason ONLY from the evidence - do not use your pre-trained knowledge
- Cite specific evidence by triple ID: [0], [3], [7]
- Explain your reasoning in 2-3 sentences
- Identify key evidence triples that support your verdict

[Few-shot examples with reasoning omitted for brevity]

Now evaluate this NEW claim:

Claim: {test_claim}

Evidence ({len(test_triples)} triples, unfiltered):
[0] {s_0} --{p_0}--> {o_0}
[1] {s_1} --{p_1}--> {o_1}
...

Output JSON (respond with ONLY valid JSON, no other text):
{
  "verdict": "SUPPORTED" or "REFUTED",
  "explanation": "Your reasoning (2-3 sentences, cite triple IDs)"
}
```

D LLM-Based Semantic Filtering Methodology

This appendix provides complete implementation details for our LLM-based semantic filtering system used to create high-quality training subsets for BERT.

D.1 Overview and motivation

FACTKG single-step subgraphs contain an average of over 50 triples per claim. When linearized for BERT, this often exceeds the 512-token limit, forcing truncation that discards potentially useful evidence. Rather than training on randomly truncated or complete dense subgraphs, we use GPT-4.1-mini as a semantic filter to identify the 10 most relevant triples per claim before BERT training.

D.2 Three-stage filtering pipeline

Our filtering pipeline consists of three stages:

Stage 1: Prefiltering (cheap lexical + heuristics). We first apply a fast prefilter that keeps up to $M = 24$ candidate triples using two criteria:

1. **Must-keep heuristics:** Triples containing predicates highly correlated with verification (e.g., `birthDate`, `spouse`, `successor`, `predecessor`) or where the subject/object appears verbatim in the claim text.

2. **Lexical scoring:** For remaining triples, compute Jaccard similarity between claim tokens and triple tokens (subject + relation + object). Rank by descending similarity.

The prefilter reduces the candidate set from $\sim$50 triples to 24, eliminating obviously irrelevant triples.

Stage 2: LLM relevance scoring. For each of the 24 prefiltered triples, we query GPT-4.1-mini to score relevance to the claim on a 0–1 scale. We use `temperature=0`, `seed=42`, and cache all scores in SQLite keyed by (`claim`, `subject`, `relation`, `object`, `model`) to ensure deterministic results and avoid redundant API calls.

Stage 3: Rank fusion. We fuse the LLM relevance scores with the lexical scores using weighted rank fusion:

$$\text{fused\_rank}(t) = \alpha\cdot\text{llm\_rank}(t)+(1-\alpha)\cdot\text{lex\_rank}(t)$$

where $\alpha = 0.7$. Lower fused rank is better. This combines semantic understanding (LLM) with surface-level alignment (lexical) to produce a final ranking. We keep the top $k = 10$ triples.

D.3 Stratified sampling for training subset

When creating the 9,645-example training subset, we use stratified sampling by reasoning type to ensure balanced coverage:

1. Explode the reasoning type list (claims can have multiple types)

2. Compute target samples per type: $n_{\text{per\_type}} = \lfloor N_{\text{total}}/N_{\text{types}} \rfloor$

3. Sample $\min(n_{\text{per\_type}}, n_{\text{available}})$ examples per type with `random_state=42`

4. Deduplicate by claim text (since one claim may appear under multiple types)

This prevents the filtered dataset from being dominated by a single reasoning type and ensures BERT sees diverse examples during training.

E Detailed Error Analysis

This appendix provides a comprehensive quantitative analysis of the 300-example LLM comparison between memorization and KG-grounded reasoning. We systematically categorize errors, quantify their occurrence, and provide annotated examples to illustrate each failure mode.

E.1 Asymmetric error patterns

Of the 300 test examples, we identified 49 cases where KG-grounded reasoning succeeded but memorization failed, and 21 cases where memorization succeeded but KG-grounding failed. This asymmetry (49 vs. 21) demonstrates that KG evidence provides a net benefit, but not universally.

KG evidence is most beneficial for: (1) *obscure entities* where the LLM has no pre-trained knowledge (e.g., “Mo Courtney had a religion”: memorization lacks the fact, but the KG provides the triple); (2) *question-type claims* where memorization refuses to answer (“What is the name of Pat Screen’s spouse?”); and (3) *formal KG relations* that differ from common-sense interpretations (“Keith Haring had a predecessor”: memorization interprets this as artistic influence, but the KG checks for a formal predecessor relation and finds none).

Conversely, KG evidence can mislead when: (1) *evidence is too dense* for complex multi-hop claims, overwhelming the model with irrelevant triples; (2) *information is implicit* (e.g., college attendance dates can be inferred from era and team affiliation, but the KG lacks explicit date triples); and (3) *relational direction is ambiguous* (e.g., "prime minister to X" vs. "prime minister of X" are easily confused).

E.2 Multi-hop puzzle and label bias

Multi-hop claims show *identical* performance (73.03%) in both memorization and KG-grounded modes, with no improvement from evidence provision. Of 89 multi-hop claims, both approaches failed on 15 and succeeded on 56, suggesting that multi-hop reasoning requires explicit chaining logic that GPT-4.1-mini struggles to perform regardless of whether evidence is provided.

The test set contains 123 SUPPORTED (41%) and 177 REFUTED (59%) claims. KG-grounded reasoning achieves 93.79% accuracy on REFUTED claims but only 69.92% on SUPPORTED, indicating that detecting absence or contradiction in the KG is easier than confirming presence. This asymmetry reflects the difficulty of confirmation versus refutation in dense KG evidence.

E.3 Citation analysis

As can be seen in Table 11, 100% of KG-grounded responses include explicit triple citations, with an average of 6.28 cited triple IDs per explanation. Since the average subgraph contains 36.7 triples across the 300 examples, this demonstrates that the model is *selectively* citing relevant evidence rather than blindly referencing all available triples. This provides confidence that the KG-grounded model is genuinely performing evidence-based reasoning rather than pattern matching on the prompt structure.

Table 11: Citation statistics in KG-grounded responses.

Metric	**Value**
Responses with citations	300 (100.0%)
Responses without citations	0 (0.0%)
Avg. cited triples per response	6.28
Median cited triples per response	4
Max cited triples in one response	59

E.4 Overall error distribution

Table 12 summarizes the distribution of correct and incorrect predictions across both conditions. The asymmetry is striking: KG-grounding rescues 49 examples where memorization failed, while memorization rescues only 21 where KG-grounding failed. This roughly 2.3:1 ratio provides strong evidence for the value of explicit evidence provision.

Table 12: Error distribution across memorization and KG-grounded conditions on 300 test examples.

Condition	**Correct**	**Accuracy**
Memorization	224	74.67%
KG-grounded	252	84.00%
Both correct	203 (67.7%)	
Both incorrect	27 (9.0%)	
Only Mem correct	21 (7.0%)	
Only KG correct	49 (16.3%)	

E.5 Paired Significance Details

Table 13 reports the full paired test statistics underlying the memorization vs. KG-grounded comparison in Section 8.2. For each model and reasoning type, we apply McNemar's exact test on the discordant pairs: b (claims where KG-grounded is correct but memorization is not) and c (the reverse). Under the null hypothesis of no difference, $\min(b, c) \sim \text{Bin}(b + c,\ 0.5)$; we report the two-sided exact p-value. Paired bootstrap 95% confidence intervals for Δ (20,000 resamples, seed 42) are reported for the overall comparison only, where the sample size of 300 warrants stable interval estimates; per-type bootstrap CIs are omitted because the smaller per-type n (70–132) yields intervals too wide to be informative. The 2:1 ratio of b to c at the overall level (63 vs. 25 for GPT-4o-mini; 49 vs. 21 for GPT-4.1-mini) confirms the net direction and magnitude of the KG-grounding benefit. At the per-type level, only existence yields individually significant improvements for both models ($p < 0.001$); negation for GPT-4.1-mini approaches significance ($p = 0.061$), and multi-hop shows identical discordant counts ($b = c = 9$, $p = 1.0$), consistent with the stagnation observed in Section 9.

Table 13: Full paired significance details. b and c are discordant pairs (KG-correct/Mem-wrong and vice versa). McNemar's test used; CIs are for Δ (bootstrapped).

		Δ	**Discordant**		
Model	**Type**	**(95% CI)**	b	c	p**-val**
GPT-4o-mini	Overall	+12.67 (6.7, 18.7)	63	25	< 0.001
	Existence	+34.29	24	0	< 0.001
	Substitution	–6.82	9	18	0.122
	Multi-hop	+10.11	19	10	0.136
	Multi-claim	–2.02	9	11	0.824
	Negation	+9.78	18	9	0.122
GPT-4.1-mini	Overall	+9.33 (4.0, 14.7)	49	21	0.001
	Existence	+22.86	18	2	< 0.001
	Substitution	+3.03	10	6	0.455
	Multi-hop	+0.00	9	9	1.000
	Multi-claim	+5.05	12	7	0.359
	Negation	+11.96	20	9	0.061