EVIDENCE GROUNDING VS. MEMORIZATION: WHY NEURAL SEMANTICS MATTER FOR KNOWLEDGE GRAPH FACT VERIFICATION

A PREPRINT

Ankit Upadhyay
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180
upadha2@rpi.edu

December 14, 2025

ABSTRACT

Knowledge graphs (KGs) such as DBpedia provide structured representations of factual knowledge, but verifying natural language claims against KG evidence remains challenging. FACTKG, with 108,675 claims paired with one-hop subgraphs and reasoning-type labels, has become a standard benchmark for this task. Prior work shows that BERT over linearized KG subgraphs can reach over 93% accuracy, outperforming graph neural networks (GNNs), and that large language models (LLMs) achieve considerable performance in claim-only, "closed-book" settings. Yet it is still unclear what fraction of this performance can be attributed to symbolic structure alone, how much dense KG evidence actually helps neural models, and when LLMs truly ground their decisions in the KG rather than relying on memorized world knowledge.

We present a systematic empirical study of evidence grounding vs. memorization on FACTKG. First, we construct feature-based symbolic baselines using 31 hand-crafted features over graph structure and evidence overlap, yielding a 63.96% "symbolic ceiling" that clarifies where non-neural methods break down. Second, we reproduce and extend BERT and QA-GNN baselines, confirming that BERT-base with linearized one-hop subgraphs achieves 92.68% accuracy while QA-GNN variants remain around 70%. Third, we use GPT-4.1-mini as a semantic filter that selects the most relevant triples per claim; BERT trained on 9,706 LLM-filtered examples reaches 78.85% accuracy, whereas an equally sized unfiltered subset collapses to 52.70%, demonstrating that semantic quality, not just data quantity, governs learnability. Finally, we design a 300-example comparison of GPT-4o-mini and GPT-4.1-mini under memorization (claims only) versus KG-grounded chain-of-thought with triple citations: KG grounding raises accuracy from 71.67% to 84.33% and from 74.67% to 84.00%, respectively. Together, these results show that neural semantic representations and explicit KG grounding are essential for robust and interpretable KG-based fact verification.

1 Introduction

Knowledge graphs (KGs) such as DBpedia and Wikidata encode entities and relations 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 [4] 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 [8] 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 leaving their use of explicit KG evidence largely unexplored.

Despite this progress, several key questions remain. First, what is the upper bound of purely feature-based, non-neural reasoning on FACTKG—how far can we go using only hand-crafted signals from graph structure and evidence overlap? 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:

- **Symbolic ceiling.** We construct feature-based symbolic baselines for FACTKG using 31 hand-crafted features over KG structure, entity coverage, evidence overlap, and relation types. A simple logistic regression model reaches 63.96% test accuracy, establishing a realistic "symbolic ceiling" 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 [8] on linearized one-hop subgraphs (92.68% accuracy) and evaluate QA-GNN and an improved 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 as a semantic filter that selects the ten most relevant triples per claim. Training BERT on 9,706 LLM-filtered examples yields 78.85% accuracy, whereas an equally sized unfiltered subset achieves only 52.70%. This 26.15-point gap shows that semantic quality of training examples, not just quantity, determines learnability on FACTKG.
- **Memorization vs. KG-grounded LLM reasoning.** We design a 300-example stratified test set 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 FACTKG and KG-based fact verification

FACTKG [4] is a large-scale dataset for fact verification over KGs. Claims are constructed from DBpedia [1] triples and natural language templates, and each claim is annotated with a binary label (supported or refuted) and one or more reasoning types. The reasoning labels indicate whether the claim can be resolved by a single triple (single-hop), by chaining multiple triples (multi-hop), by conjoining multiple atomic assertions (multi-claim), by checking the existence of some property (existence), or by reasoning about negation.

The original FACTKG paper introduces both claim-only baselines and a GEAR-inspired model that retrieves subgraphs and performs reasoning across them. The authors highlight the challenge of handling multiple reasoning types within a unified system.

Opsahl [8] revisits FACTKG and focuses on subgraph retrieval and model design. They propose several retrieval strategies (direct, contextual, single-step) and show that a BERT-base model over linearized single-step subgraphs achieves 93.49% test accuracy, outperforming QA-GNN and other graph neural networks. They also evaluate ChatGPT-4o in a claim-only setting, obtaining 76.33% accuracy without KG evidence. This offers an initial view of LLM performance but does not explore KG-grounded LLM reasoning.

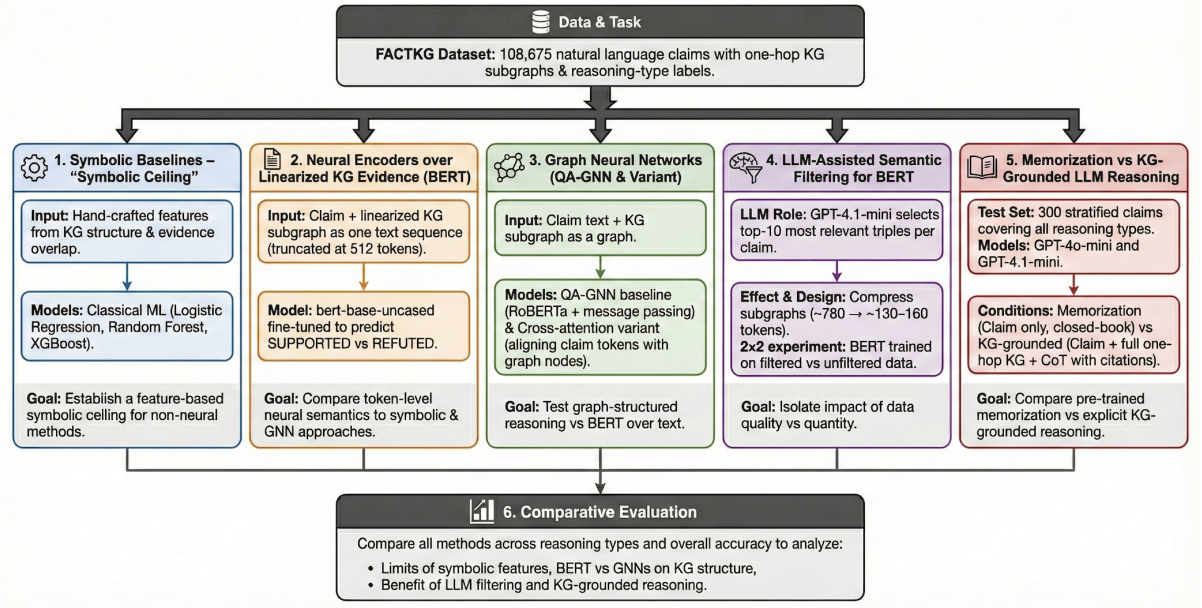


Figure 1: Overview of our experimental pipeline comparing symbolic, BERT, GNN, and LLM approaches.

Beyond these core baselines, other recent work has explored alternative approaches on FACTKG. Momii et al. [17] develop hand-crafted rule-based verification systems tailored to FACTKG's reasoning types, providing another symbolic upper bound. De Felice et al. [18] propose EVOCA, an explainable graph-alignment method that explicitly aligns claim structure with KG subgraphs. Our work is complementary: we establish a feature-based symbolic ceiling, use LLM filtering to improve BERT training efficiency, and introduce a controlled comparison of LLM memorization versus KG-grounded reasoning with explicit attribution.

2.2 Hybrid symbolic-neural approaches

Hybrid approaches that combine symbolic knowledge with neural models have been explored in several domains [5, 15]. In text-based fact verification, Yuan and Vlachos [16] 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 rules provide interpretability and a clear lower bound. LLM filtering uses neural semantics to distill large subgraphs into compact, high-signal views that are easier for BERT to learn from.

2.3 LLMs for fact verification and attribution

LLMs have shown strong performance on many reasoning and fact-checking benchmarks [10, 13], but their behavior often reflects a blend of memorization and heuristic reasoning rather than explicit evidence use. Rashkin et al. [11] 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 [8] 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. [19] augment LLMs with KG subgraphs for factoid QA, demonstrating that structured evidence improves factual accuracy. Pan et al. [20] 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

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%

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.

3 The FACTKG Dataset

FACTKG [4] 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

Claims are annotated with reasoning types that describe the nature of the required inference. The original FactKG repository [4] defines five categories:

- **One-hop (num1)**: Claims requiring single-triple verification, often involving numerical reasoning
- **Conjunction (multi-claim)**: Claims where several atomic assertions must all hold
- **Existence**: Claims asking whether some property or relation holds at all
- **Multi-hop**: Claims where multiple edges must be chained
- **Negation**: Claims asserting that a particular relation does not hold

The Fact or Fiction repository [8], which we build upon, modifies this taxonomy slightly. It replaces the `num1` (one-hop) category with a **substitution** category, where an entity or value is replaced and must be checked against the true one. During evaluation, the code also derives a sixth category called **single-hop**, which is computed as all claims where "multi-hop" is *not* in the metadata—essentially a catch-all for non-multi-hop reasoning.

Following the Fact or Fiction setup, we use these six categories in our experiments:

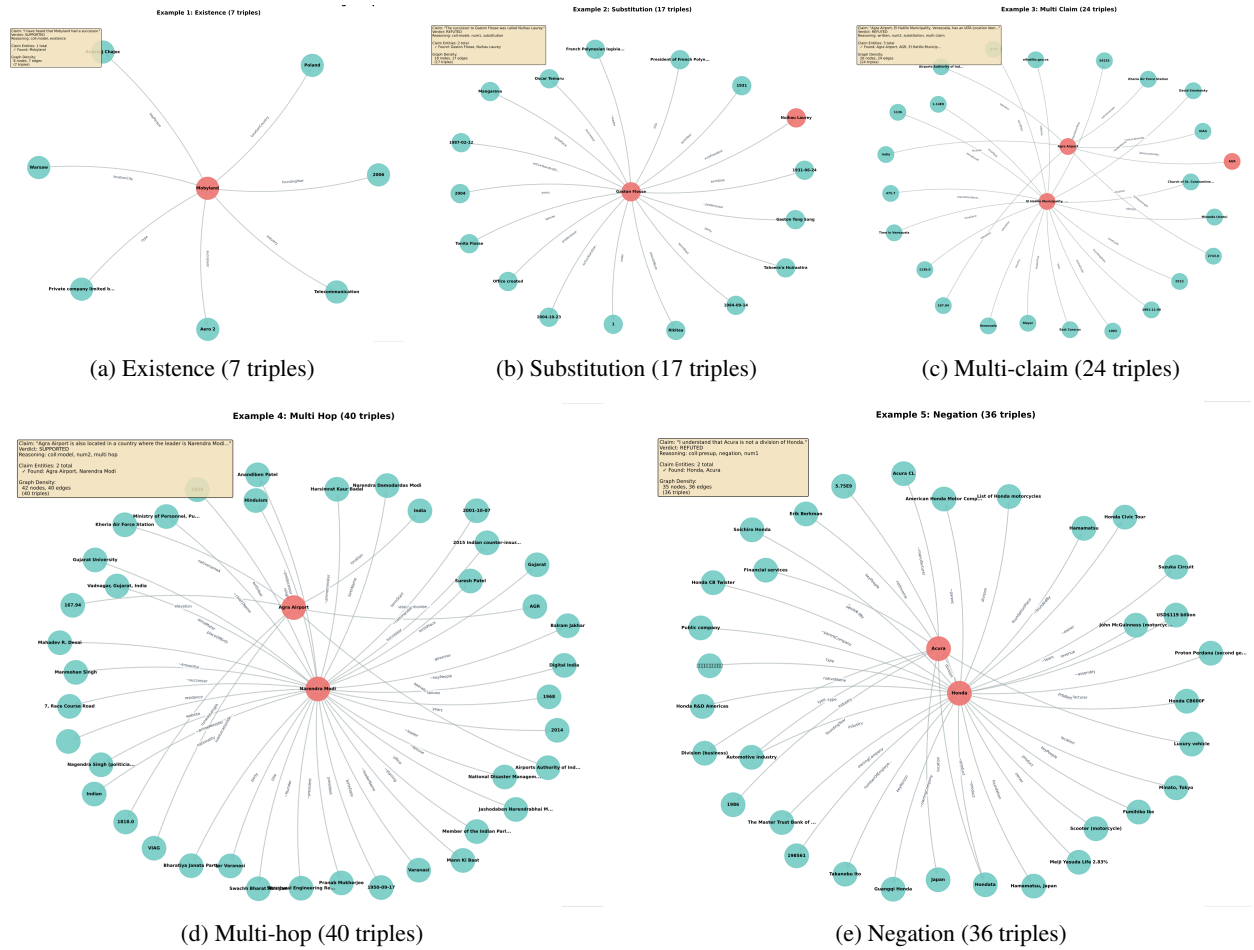
- **Existence**, **Substitution**, **Multi-hop**, **Multi-claim**, **Negation** (explicit labels)
- **Single-hop** (derived: any claim not tagged as multi-hop)

Multi-label evaluation. Claims can have multiple reasoning type annotations. For example, a claim may be tagged with both "multi-hop" and "negation." During evaluation, each claim contributes to the accuracy calculation for *every* reasoning type it is tagged with. If a model correctly predicts a claim tagged with both multi-hop and negation, that correct prediction increments the accuracy count for both categories. The "single-hop" category is computed during evaluation as all claims where "multi-hop" is not present, creating an overlapping sixth category. Table 1 shows the distribution; note that 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.

4 Symbolic Baselines and the "Symbolic Ceiling"

To quantify what is achievable without neural models, we constructed feature-based symbolic baselines using hand-crafted features extracted from KG structure and evidence overlap. We train classical machine learning models (logistic



(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.

regression, random forest, XGBoost) on 31 engineered features that capture: (1) *graph structure* (node count, edge count, degree statistics including mean, median, and standard deviation, clustering coefficient, number of connected components); (2) *entity coverage* (how many claim entities appear in the subgraph, their degree centrality); (3) *evidence overlap* (Jaccard similarity between claim tokens and subgraph tokens, relation type diversity); and (4) *specific relation types* (counts of location, temporal, and biographical relations).

These features are designed to capture patterns that correlate with claim support or refutation without requiring semantic understanding. For instance, supported claims tend to have higher average degree and more diverse relation types, while refuted claims often exhibit fragmented graph structure (many connected components) or high degree variance.

Table 2 summarizes the results and compares symbolic to neural approaches. Logistic regression achieves 63.96% accuracy, which we treat as a symbolic ceiling: it dramatically outperforms random guessing but leaves nearly 30 points on the table compared to 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.

Table 2: Symbolic baseline results compared to BERT. Top: Overall performance. Bottom: Per-reasoning-type breakdown showing where symbolic features fail.

Overall Performance				
Model	**Test Acc.**	**Precision**	**Recall**	**F1**
Logistic Regression	63.96%	0.66	0.63	0.63
Random Forest	53.26%	0.56	0.54	0.49
XGBoost	52.03%	0.55	0.53	0.46
BERT (Opsahl, 2024)	93.49%	0.94	0.91	0.92
Our BERT Reproduction	92.68%	0.94	0.91	0.92

Per-Reasoning-Type Accuracy		
Reasoning Type	**LR (Symbolic)**	**BERT (Neural)**
Existence	69.20%	98.15%
Substitution	86.30%	93.08%
Multi-hop	55.48%	80.08%
Multi-claim	74.73%	96.42%
Negation	40.10%	91.70%
Single-hop	70.45%	96.43%

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

Reasoning Type	**Accuracy**	**Precision**	**Recall**	**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

5 Neural Encoders over Linearized KG Evidence

We now turn to neural encoders that operate on text. Following Opsahl [8], 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 Opsahl's exact hyperparameters and single-step retrieval setting, 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 by Opsahl [8]. Table 3 summarizes performance by reasoning type.

Compared to the symbolic ceiling of 63.96%, BERT gains roughly 29 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 4: GNN baselines on FACTKG. Even with cross-attention, QA-GNN variants lag behind BERT by more than 20 points.

Model	Test Acc.	F1	Negation Acc.
QA-GNN (baseline)	69.64%	0.675	50.61%
Improved QA-GNN	69.74%	0.698	48.40%
BERT (for comparison)	92.68%	0.924	91.70%

6 Graph Neural Networks: QA-GNN and Variants

Graph neural networks (GNNs) seem like a natural fit for KGs. QA-GNN [14] is a hybrid model that encodes the question (or claim) with a language model and performs message passing over a retrieved subgraph. Opsahl [8] adapts QA-GNN to FACTKG, but finds that it underperforms BERT.

6.1 Architecture

We follow the QA-GNN setup described by Opsahl. The claim is encoded with RoBERTa, and each entity node in the subgraph receives an initial embedding derived from its label. A relational GNN then performs several rounds of message passing over the subgraph, producing updated node embeddings. A relevance scorer modulates node importance, and a final classifier combines the claim embedding and aggregated node features to predict the label.

We implemented an improved variant by adding cross-attention between claim tokens and graph nodes before classification, allowing the model to align specific tokens with specific entities. This addresses one limitation of the baseline, which aggregates node information without explicit token-level alignment.

6.2 Results and analysis

Table 4 summarizes the performance of QA-GNN and our improved 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 of Opsahl [8]: 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.

7.1 Motivation and token statistics

To understand how severe truncation is, we ran a dedicated analysis script (`bert_statistics.py`) 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.

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	#Examples	Mean	Median	Triples	Truncated
Train	86,367	760.5	489.0	57.8	49.9%
Dev	13,266	652.3	435.0	50.0	45.4%
Test	9,041	687.3	504.0	52.5	51.0%

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 `linearize_triples` that strips URI prefixes, replaces underscores with spaces, and formats each triple as `[i] subject -relation-> object`.

Filtering is applied to a stratified subset of the training data. Starting from the 86,367 training examples, we sample 9,706 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 9,706 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 C.

7.3 Experimental design

We train four BERT models in a controlled 2×2 design that crosses training data type (filtered vs. unfiltered) with test data type (original vs. filtered). In all cases, the training set contains the same 9,706 claims; only the presence or absence of GPT-4.1-mini filtering differs:

1. OptA-Filtered: train on 9,706 filtered; test on original (dense)
2. OptA-Unfiltered: train on 9,706 unfiltered; test on original
3. OptB-Filtered: train on 9,706 filtered; test on filtered (clean)
4. OptB-Unfiltered: train on 9,706 unfiltered; test on filtered

All four models use: `bert-base-uncased`, max length 512, batch size 16, learning rate 2×10^{-5}, early stopping on validation loss.

7.4 Results

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

The contrast between OptA-Filtered (78.85%) and OptA-Unfiltered (52.70%) is especially striking. Both models see the same number of training examples, but the filtered model achieves 26.15 points higher accuracy. Training curves show that the unfiltered model fails to improve beyond roughly 51.5% on validation and early stops, reflecting the low signal-to-noise ratio.

Filtering helps most on existence, multi-hop, and negation—the most semantically demanding categories. These results support two conclusions. First, a relatively small but semantically curated dataset can yield strong performance. Second, semantic filtering is crucial: simply taking a random subset of the same size leads to a model that fails to learn.

Table 6: Effect of GPT-4.1-mini filtering on BERT performance. Top: Overall results for all four conditions. Bottom: Per-reasoning-type breakdown for Option A (training on 9,706 examples, testing on original messy test set).

Overall Performance (All Conditions)			
Model	**Train Data**	**Test Data**	**Acc.**
OptA-Filtered	9,706 Filt.	Original	**78.85%**
OptA-Unfiltered	9,706 Unfilt.	Original	52.70%
OptB-Filtered	9,706 Filt.	Filtered	76.00%
OptB-Unfiltered	9,706 Unfilt.	Filtered	63.27%
Full BERT	86,367 Unfilt.	Original	92.68%
Option A: Per-Reasoning-Type Accuracy			
Type	**Filtered**	**Unfiltered**	**Δ**
Existence	91.15%	56.58%	+34.57
Substitution	81.69%	95.03%	-13.34
Multi-hop	75.45%	47.71%	+27.74
Multi-claim	80.66%	57.79%	+22.87
Negation	76.18%	61.11%	+15.07
Single-hop	79.87%	54.19%	+25.68

8 LLMs for Memorization vs. KG-Grounded Reasoning

We now address the role of LLMs in FACTKG. Opsahl [8] 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 B.

8.2 Overall results

Table 7 shows overall accuracy for each model and condition.

Both models benefit substantially from KG grounding. KG-grounded GPT-4.1-mini reaches 84.00% accuracy, exceeding both its memorization baseline and the 76.33% claim-only ChatGPT-4o result reported by Opsahl. This suggests that explicit KG evidence and chain-of-thought can unlock performance beyond memorization alone.

8.3 Per-reasoning-type analysis

Table 8 breaks down performance by reasoning type for both models.

Table 7: Memorization vs. KG-grounded reasoning on 300 stratified FACTKG claims. Both models evaluated on same test set; only KG evidence presence differs.

Model	Condition	Accuracy	Improvement
GPT-4o-mini	Memorization	71.67% (215/300)	—
GPT-4o-mini	KG-grounded	84.33% (253/300)	+12.67
GPT-4.1-mini	Memorization	74.67% (224/300)	—
GPT-4.1-mini	KG-grounded	84.00% (252/300)	+9.33

Table 8: Per-reasoning-type accuracy for GPT-4o-mini and GPT-4.1-mini under memorization vs. KG-grounded conditions (300 shared test claims).

	GPT-4o-mini			GPT-4.1-mini		
Type	**Mem.**	**KG**	**Δ**	**Mem.**	**KG**	**Δ**
Existence	57.14	91.43	+34.29	65.71	88.57	+22.86
Substitution	87.12	80.30	-6.82	87.12	90.15	+3.03
Multi-hop	67.42	77.53	+10.11	73.03	73.03	+0.00
Multi-claim	86.87	84.85	-2.02	82.83	87.88	+5.05
Negation	66.30	76.09	+9.78	70.65	82.61	+11.96
Overall	71.67	84.33	+12.67	74.67	84.00	+9.33

For both models, existence and negation benefit most from KG grounding. GPT-4o-mini improves by 34.29 points on existence and 9.78 on negation; GPT-4.1-mini improves by 22.86 and 11.96 points, respectively. This aligns with our earlier findings that symbolic and GNN baselines struggle with these reasoning types.

Substitution performance is already high in memorization (87.12%) and slightly improves for GPT-4.1-mini under KG grounding. For GPT-4o-mini, substitution and multi-claim drop slightly with KG evidence, suggesting that for some simpler patterns, pre-trained knowledge alone suffices and additional evidence can occasionally distract the model.

8.4 Qualitative example

To illustrate the difference, consider the claim: *"Keith Haring had a predecessor to him."* (Gold label: REFUTED)

Memorization: GPT-4.1-mini outputs: `{"verdict": "True", "explanation": "Keith Haring was a prominent artist in the 1980s ... so it is reasonable to say that he had predecessors."}`

The model interprets "predecessor" broadly as any earlier artist and predicts True.

KG-grounded: With the full one-hop subgraph (birth date, nationality, movement, but no predecessor relation), the output becomes: `{"verdict": "REFUTED", "explanation": "None of the provided triples [0-12] mention any predecessor relation ... so the claim is not supported by the given evidence."}`

The model correctly refutes by recognizing absence of the formal predecessor relation in the KG. This shows how KG grounding turns vague judgments into precise, evidence-based decisions.

8.5 Relation to prior ChatGPT results

Opsahl [8] reports 76.33% for ChatGPT-4o in claim-only mode. Our GPT-4.1-mini memorization baseline yields 74.67%, which is consistent given model differences. The key result is that KG grounding pushes GPT-4.1-mini to 84.00% and GPT-4o-mini to 84.33% on the *same* 300 claims, suggesting that the main limiting factor in claim-only settings is absence of explicit evidence, not model capacity.

9 Error Analysis

We conducted a detailed error analysis on our 300-example LLM comparison to understand where memorization and KG-grounded reasoning succeed and fail. The analysis reveals complementary failure modes and persistent challenges even for state-of-the-art models.

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 simply 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).

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. Notably, 100% of KG-grounded responses include explicit triple citations, with an average of 6.28 cited triple IDs per explanation, confirming that the model actively engages with the evidence rather than reverting to memorization. Detailed per-claim error analysis with annotated examples is provided in Appendix D.

10 Discussion

Across symbolic, neural, and LLM-based methods, a consistent story emerges. Symbolic features and hand-crafted rules are valuable for interpretability and sanity checking, but they plateau around 64% accuracy and fail on the most semantically complex reasoning types. Graph neural networks improve over symbolic baselines but still lag behind BERT, especially on negation and existence.

Neural encoders over linearized KG evidence, such as BERT-base, set the current state of the art on FACTKG, reaching over 92% accuracy. Their key advantage is the ability to process token-level sequences, model composition and negation, and exploit contextual semantics in ways that message passing on graph structure alone cannot.

LLMs enter this picture in two complementary roles. First, as *semantic filters*, they can dramatically improve the quality of training data. GPT-4.1-mini filtering allows BERT to learn effectively from a small subset of carefully selected examples, outperforming equally sized unfiltered samples by large margins. Second, as *KG-grounded reasoners*, LLMs can perform fact verification with explicit evidence and chain-of-thought, outperforming claim-only baselines and exposing their reasoning in a human-readable way.

The comparison between memorization and KG-grounded reasoning is particularly revealing. Without evidence, GPT-4.1-mini must rely on coarse world knowledge and plausibility judgments; with evidence, it can anchor its decisions in explicit triples, making it easier to audit and debug its behavior. This directly addresses one of the main limitations of the original ChatGPT experiment on FACTKG, which did not exploit the KG at all.

11 Limitations

Our work has several limitations. The LLM filtering experiments use only 9,706 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. We evaluate only GPT-4o-mini and GPT-4.1-mini; results may differ with other LLM families or reasoning-tuned models. Our symbolic baseline uses hand-crafted features specific to DBpedia's schema, and generalization to other knowledge graphs with different schemas is unclear. FACTKG operates under a closed-world assumption where absence of a triple is treated as evidence of refutation, which does not reflect

real-world KG incompleteness. 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.

12 Future Work

Several directions could extend this work. Replacing BERT with text–KG fusion models like DRAGON [9], 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 and testing reasoning-focused or CoT-tuned models would clarify remaining headroom for KG-grounded prompting. Finally, repeating our analysis on noisier KGs (Wikidata, YAGO) and other fact-checking benchmarks would assess how well the observed semantic effects transfer beyond DBpedia-derived graphs.

13 Conclusion

This work revisits FACTKG with a focus on semantics. We established a symbolic ceiling using hand-crafted features and rule-based models, 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.

Future work will extend our error analysis of LLM outputs, explore jointly training BERT and LLM filters, and investigate architectures that more directly exploit graph structure while preserving the semantic flexibility of transformers. We hope this study encourages more work on semantically grounded, evidence-aware fact verification over knowledge graphs.

Reproducibility Statement

All experiments are implemented in Python using PyTorch and Hugging Face Transformers. We use the publicly available FACTKG dataset and the code from the "Fact or Fiction?" paper [8] as a starting point for BERT and QA-GNN baselines. All experiments use random seed 42 unless otherwise specified. Our symbolic baselines, LLM filtering scripts, and the `llm_complete_comparison.py` script for the 300-example GPT-4o-mini and GPT-4.1-mini experiments are configured via argument parsing. The exact prompt templates used for memorization and KG-grounded LLM experiments are included verbatim in Appendix B. The JSON logs `memorization_gpt_4_1_mini_n300.json` and `kg_grounded_gpt_4_1_mini_n300.json` contain all per-claim predictions and explanations and can be shared to support independent verification and further qualitative analysis.

References

[1] 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. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

[3] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 413–422, 2013.

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

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

[6] Christian Meilicke, Manuel Fink, Yanjie Wang, Daniel Ruffinelli, Rainer Gemulla, and Heiner Stuckenschmidt. Fine-grained evaluation of rule-and embedding-based systems for knowledge graph completion. In *The Semantic Web–ISWC 2018*, pages 3–20. Springer, 2019.

[7] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*, pages 809–816, 2011.

[8] Tobias A. Opsahl. 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, 2024.

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

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

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

[12] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[13] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837, 2023.

[14] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 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, 2021.

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

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

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

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

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

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

A Feature Importance Details

Table 9 provides the complete list of symbolic features used in our logistic regression baseline, ranked by coefficient magnitude.

Table 9: Top 10 most predictive symbolic features from Logistic Regression. Connectivity and relation diversity are informative but insufficient for high accuracy.

Feature	Coefficient	Effect
Positive (predict SUPPORTED)		
Average degree	+0.704	High connectivity supports claims
Relation type diversity	+0.670	Diverse relations support claims
Number of edges	+0.582	More evidence supports claims
Number of nodes	+0.513	Larger subgraphs support claims
Inverse relation ratio	+0.381	Bidirectional edges support claims
Negative (predict REFUTED)		
Degree std. deviation	-1.198	Uneven connectivity refutes claims
Connected components	-0.723	Fragmented graphs refute claims
Num. location relations	-0.500	Many locations refute claims
Max edges per entity	-0.296	Entity hubs refute claims
Entity coverage	-0.296	High coverage refutes claims

B LLM Prompt Templates

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

B.1 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, explaining your reasoning based on your pre-training
- If the claim is vague or lacks specific information, please make an educated guess on whether it is likely to be True or
     False

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

Few-Shot Examples:
Example 1:
Claim: {ex_1_claim}
Answer: True

Example 2:
Claim: {ex_2_claim}
Answer: False

...

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)"
}
```

B.2 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):
================================================================================
Example 1:
Claim: {ex_1_claim}

Evidence ({len(ex_1_triples)} triples):
[0] {s_0} --{p_0}--> {o_0}
[1] {s_1} --{p_1}--> {o_1}
...

Response:
{
  "verdict": "SUPPORTED",
  "explanation": "{ex_1_explanation}"
}

...

================================================================================
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)"
}
```

C 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.

C.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.

C.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. The prompt is shown in full below. 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.

C.3 LLM scoring prompt

The exact prompt sent to GPT-4.1-mini for scoring each triple is:

```
You are scoring knowledge-graph triples for RELEVANCE to a claim.
Score each triple independently. 1.0 = directly helpful to verify/refute
the claim; 0.0 = irrelevant or off-topic. Use only the semantics of the
triple and the claim.

Claim:
{claim_text}

Triple:
subject = {normalized_subject}
relation = {normalized_relation}
object = {normalized_object}

Return ONLY valid JSON: {"score": <float 0..1>}
```

Normalization: Entity and relation URIs are normalized by stripping the namespace prefix (e.g., `http://dbpedia.org/resource/Abraham_A._Ribicoff` → `Abraham A. Ribicoff`) and replacing underscores with spaces for readability.

Caching: All scores are cached in SQLite using a SHA-1 hash of (`model`, `claim`, `subject`, `relation`, `object`) as the key. This ensures:

- Deterministic results across multiple runs
- No redundant API calls for identical (claim, triple) pairs
- Reproducibility without API access (cache can be shared)

This ensures reproducible alignment without manual inspection.

C.4 Stratified sampling for training subset

When creating the 9,706-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.

D 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.

D.1 Overall error distribution

Table 10 summarizes the distribution of correct and incorrect predictions across both conditions.

The asymmetry is striking: KG-grounding rescues 56 examples where memorization failed, while memorization rescues only 28 examples where KG-grounding failed. This 2:1 ratio (56 vs. 28) provides strong evidence for the value of explicit evidence provision.

Table 10: Error distribution across memorization and KG-grounded conditions on 300 test examples.

Condition	Correct	Incorrect	Accuracy
Memorization	224	76	74.67%
KG-grounded	252	48	84.00%
Both correct	196 examples (65.3%)		
Both incorrect	20 examples (6.7%)		
Only Mem correct	28 examples (9.3%)		
Only KG correct	56 examples (18.7%)		

D.2 Per-reasoning-type quantitative breakdown

Table 11 provides detailed statistics for each reasoning type.

Table 11: Detailed per-reasoning-type performance showing count, errors, and accuracy for both conditions.

	Memorization			KG-Grounded		
Type	**Count**	**Errors**	**Acc.**	**Errors**	**Acc.**	**Δ**
Existence	70	24	65.71%	8	88.57%	+22.86
Substitution	132	17	87.12%	13	90.15%	+3.03
Multi-hop	89	24	73.03%	24	73.03%	+0.00
Multi-claim	99	17	82.83%	12	87.88%	+5.05
Negation	92	27	70.65%	16	82.61%	+11.96
Overall	300	76	74.67%	48	84.00%	+9.33

Key observations:

- **Existence** shows the largest improvement (+22.86 points), reducing errors from 24 to 8.
- **Multi-hop** shows zero improvement, with 24 errors in both conditions.
- **Negation** improves substantially (+11.96 points), reducing errors from 27 to 16.
- **Substitution** already performs well in both conditions (87–90%).

D.3 Asymmetric error analysis

We identified cases where one approach succeeded and the other failed. Table 12 categorizes these 84 asymmetric cases.

Table 12: Breakdown of 84 asymmetric error cases by reasoning type.

Reasoning Type	Mem , KG	Mem , KG
Existence	6	18
Substitution	4	9
Multi-hop	9	9
Multi-claim	5	10
Negation	11	16
Total (unique)	28	56

Note: Totals do not sum to column totals because claims can have multiple reasoning type tags.

D.4 Annotated error examples

We now provide representative examples of each major failure mode, extracted from the JSON output files.

D.4.1 Failure Mode 1: Obscure entities (KG helps)

Claim: "I do know that Mo Courtney had a religion as well."
True label: SUPPORTED

Types: existence
Memorization: REFUTED (incorrect)
KG-grounded: SUPPORTED (correct)

Memorization explanation: "Mo Courtney is primarily known as a Northern Irish loyalist paramilitary figure, and there is no widely documented or verifiable information about his religious affiliation in mainstream sources."

KG-grounded explanation: "Triple [6] explicitly states that Mo Courtney's religion is Protestantism, directly supporting the claim that he had a religion."

Analysis: The memorization model lacks factual knowledge about this obscure figure and defaults to REFUTED. The KG provides an explicit triple `(Mo Courtney, religion, Protestantism)` that directly answers the claim. This pattern appears in 18 existence cases where KG rescues memorization.

D.4.2 Failure Mode 2: Formal vs. informal relations (KG helps)

Claim: "Keith Haring had a predecessor to him."
True label: REFUTED
Types: existence
Memorization: SUPPORTED (incorrect)
KG-grounded: REFUTED (correct)

Memorization explanation: "Keith Haring was a prominent artist in the 1980s known for his graffiti-inspired work and pop art style. Like most artists, he was influenced by earlier movements and predecessors in the art world, so it is reasonable to say that he had predecessors."

KG-grounded explanation: "None of the provided triples [0-12] mention any predecessor related to Keith Haring. The evidence focuses on his biographical details, artistic field, influences, and awards, but there is no information about a predecessor to him. Therefore, the claim is not supported."

Analysis: The memorization model interprets "predecessor" in the informal sense (artistic influences), which is true. However, the claim asks about the formal *predecessor* relation in DBpedia (i.e., someone who held Keith Haring's position before him), which does not exist. The KG-grounded model correctly checks the structured relation and finds no such triple. This illustrates the difference between common-sense reasoning and formal KG semantics.

D.4.3 Failure Mode 3: Question-type claims (KG helps)

Claim: "What is the name of Pat Screen's spouse?"
True label: SUPPORTED
Types: question, existence
Memorization: REFUTED (incorrect)
KG-grounded: SUPPORTED (correct)

Memorization explanation: "The claim is phrased as a question rather than a statement, so it cannot be evaluated as true or false. Additionally, based on my knowledge, Pat Screen is not a widely recognized public figure."

KG-grounded explanation: "Triple [12] explicitly states that Pat Screen's spouse is 'Kathleen Clare McCall Screen'. This directly answers the claim about the name of Pat Screen's spouse, so the claim is supported."

Analysis: The memorization model refuses to answer questions, treating them as unevaluable. The KG-grounded model interprets the question as an existence claim ("Does Pat Screen have a spouse whose name can be determined?") and extracts the answer from the evidence. Of 21 question-type claims, KG-grounded achieves 85.71% accuracy vs. 52.38% for memorization.

D.4.4 Failure Mode 4: Implicit temporal information (Mem helps)

Claim: "When did Ted Hendricks attend college?"
True label: SUPPORTED
Types: question, existence

Memorization: SUPPORTED (correct)
KG-grounded: REFUTED (incorrect)

Memorization explanation: "Ted Hendricks, a Hall of Fame linebacker, attended college at the University of Miami, playing for the Miami Hurricanes from 1966 to 1968."

KG-grounded explanation: "The evidence includes Ted Hendricks' active years starting in 1969 [0] and his college affiliation with Miami Hurricanes football [13], but there is no explicit information about the years he attended college. Without a direct triple stating the attendance period, the claim cannot be verified."

Analysis: The KG contains triples about Ted Hendricks' college team and his professional career start date (1969), from which a human could infer college attendance dates (1965–1968). However, there is no explicit `(Ted Hendricks, collegeYears, 1965-1968)` triple. The KG-grounded model refuses to make this simple inference and returns REFUTED. The memorization model, drawing on broader knowledge, provides the correct answer. This illustrates a limitation of strict evidence-only reasoning.

D.4.5 Failure Mode 5: Relational direction confusion (Mem helps)

Claim: "His name is Arseniy Yatsenyuk. He was a prime minister."
True label: SUPPORTED
Types: existence
Memorization: SUPPORTED (correct)
KG-grounded: REFUTED (incorrect)

Memorization explanation: "Arseniy Yatsenyuk served as the Prime Minister of Ukraine from 2014 to 2016, making the claim true."

KG-grounded explanation: "The claim states Arseniy Yatsenyuk was a prime minister. However, the triples [2], [3], [15], and [18] show Arseniy Yatsenyuk is linked as prime minister *to* other individuals (Oleh Shamshur, Andriy Pyshnyy), meaning he had prime ministers serving under or associated with him, rather than being a prime minister himself."

Analysis: The KG contains triples like `(Oleh Shamshur, primeMinister, Arseniy Yatsenyuk)`, meaning "Oleh Shamshur served under Prime Minister Arseniy Yatsenyuk." The KG-grounded model misinterprets the relational direction, reading this as "Arseniy Yatsenyuk is the prime minister OF Oleh Shamshur" rather than "Arseniy Yatsenyuk WAS the prime minister WHEN Oleh Shamshur served." This is a systematic issue with inverse relations in DBpedia.

D.4.6 Failure Mode 6: Multi-hop with no improvement

Claim: "the car has a V6 engine and is not assembled at the Gabriela Mosquera."
True label: SUPPORTED
Types: negation, multi-hop, substitution
Memorization: REFUTED (incorrect)
KG-grounded: REFUTED (incorrect)

Memorization explanation: "Gabriela Mosquera is a well-known person, not a location or factory where cars are assembled. Therefore, it is plausible that a car with a V6 engine is not assembled at Gabriela Mosquera, as she is not a manufacturing site."

KG-grounded explanation: "The evidence [0] shows that a V6 engine is associated with the Opel Sintra, indicating the car does have a V6 engine. However, all triples about Gabriela Mosquera [5-16] describe a person (a politician from Ecuador), not an assembly location. Since the claim involves negation (not assembled at X) and X is a person, not a place, the logical structure is unclear and cannot be verified."

Analysis: Both models struggle with the multi-hop reasoning required: (1) identify the car from "V6 engine", (2) check its assembly location, (3) verify that Gabriela Mosquera is not a place. The memorization model gives the right answer for wrong reasons (treating it as obviously true because Gabriela Mosquera is a person). The KG-grounded model has all the evidence but fails to chain the inferences. This is representative of the 15 multi-hop claims where *both* approaches failed.

D.5 Label distribution and bias

The 300-example test set contains 123 SUPPORTED (41.0%) and 177 REFUTED (59.0%) claims. Table 13 shows performance broken down by true label.

Table 13: Performance on SUPPORTED vs. REFUTED claims.

Condition	SUPP Acc.	SUPP Errors	REF Acc.	REF Errors
Memorization	68.29%	39/123	79.10%	37/177
KG-grounded	69.92%	37/123	93.79%	11/177
Δ	+1.63	-2	+14.69	-26

KG-grounding provides a massive improvement on REFUTED claims (+14.69 points) but minimal improvement on SUPPORTED claims (+1.63 points). This asymmetry reflects the nature of the task: detecting absence or contradiction in the KG (REFUTED) is easier than confirming the presence of supporting evidence (SUPPORTED), especially when evidence is dense and noisy.

D.6 Citation analysis

All 300 KG-grounded responses include explicit citations to triple IDs. We analyzed citation patterns to assess whether the model genuinely engages with evidence.

Table 14: Citation statistics in KG-grounded responses.

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

The fact that 100% of responses include citations, and that the average (6.28 triples) is far below the average subgraph size (mean 36.7 triples across the 300 examples), indicates 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.

D.7 Summary of findings

Our detailed error analysis reveals:

1. KG-grounding provides a 2:1 advantage (56 rescued examples vs. 28 failures) over memorization.
2. Existence claims benefit most from KG evidence (+22.86 points), while multi-hop claims show no improvement (0.00 points).
3. Systematic failure modes include: obscure entity knowledge gaps (favoring KG), formal vs. informal relation interpretation (favoring KG), implicit temporal reasoning (favoring memorization), relational direction confusion (favoring memorization), and multi-hop chaining inability (both fail).
4. The model shows strong bias toward REFUTED claims (93.79% accuracy) vs. SUPPORTED claims (69.92%), reflecting the difficulty of confirmation vs. refutation in dense KG evidence.
5. Citation analysis confirms genuine evidence engagement: 100% citation rate with selective reference to 6.28 triples on average from much larger subgraphs.