Phase 2 Evaluation Report

Personal Research Portal : Carbon Footprint of LLMs

Group 4: Dhiksha Rathis, Shreya Verma

Generated: 2026-02-15

1. System Overview

The Personal Research Portal investigates the carbon footprint of large language models (LLMs). The system ingests a corpus of 20 sources (14 peer-reviewed papers, 3 technical reports, and 3 tools/workshop papers) and uses a Retrieval-Augmented Generation (RAG) pipeline to answer research queries with citation-backed evidence.

Technical stack: Documents are chunked using a section-aware sliding window (500 tokens, 100-token overlap) and embedded with *all-MiniLM-L6-v2* (384-dim). Vectors are indexed in FAISS (IndexFlatIP with cosine similarity). Generation uses Grok-3 as the primary LLM. The baseline retrieves top-5 chunks via semantic search; the enhanced pipeline adds query rewriting and sub-query decomposition, merging top-8 chunks. An LLM-as-Judge approach (temperature=0.0, structured JSON) scores outputs.

2. Query Set Design

A 20-query evaluation set tests three dimensions of retrieval quality:

Category	Count	Purpose
Direct (D01–D10)	10	Single-source factual retrieval, tests basic grounding and citation accuracy
Synthesis (S01–S05)	5	Cross-source comparison, tests multi-source integration and reasoning
Edge Case (E01–E05)	5	Corpus boundary detection, tests trust, uncertainty handling, and refusal to hallucinate

Direct queries target specific claims from individual papers (e.g., CO2 emissions for BERT, tools for carbon tracking). Synthesis queries require comparing multiple sources (e.g., Strubell vs. Patterson methodology differences). Edge cases probe out-of-corpus topics (GPT-4 footprint, quantum computing, carbon offsets) where the system must correctly flag missing evidence.

3. Evaluation Metrics

Six metrics evaluate RAG quality across complementary dimensions. Three are LLM-judged on a 1–4 scale (Groundedness, Answer Relevance, Context Precision), two are deterministic (Citation Precision as valid/total citations, Source Recall as expected sources found/total expected), and one is rule-based (Uncertainty Handling: Y/N for flagging missing evidence). Pass/warn/fail thresholds are set at ≥3.5 / ≥2.5 / <2.5 respectively for the LLM-judged metrics.

4. Results

4.1 Baseline RAG : Aggregate Scores

Type	n	Ground.	Relev.	Cite Prec.	Src Recall
Direct	10	2.60	3.60	0.97	0.42
Synthesis	4	2.75	2.00	1.00	0.31
Multihop	1	4.00	4.00	1.00	0.33
Edge Case	5	3.80	2.80	1.00	0.25
Overall Avg	**20**	**3.00**	**3.10**	**0.99**	**0.37**

The baseline shows strong citation precision (0.99) but critically low source recall (0.37), meaning the system cites correctly from what it retrieves, but frequently fails to retrieve the expected sources. Groundedness averages 3.0 (borderline warn), with direct queries weakest at 2.6. Edge cases perform well on groundedness (3.8) because the system generally flags missing evidence rather than hallucinating. Synthesis queries struggle on relevance (2.0), often failing to retrieve enough sources for meaningful comparisons. Uncertainty flagging appeared in 8 of 20 baseline runs.

4.2 Enhanced RAG : Aggregate Scores

Type	n	Ground.	Relev.	Cite Prec.	Src Recall
Direct	10	2.60	3.60	0.97	0.42
Synthesis	4	2.75	3.50	1.00	0.44
Multihop	1	4.00	4.00	0.93	0.33
Edge Case	5	3.60	2.80	1.00	0.25
Overall Avg	**20**	**2.95**	**3.40**	**0.98**	**0.40**

4.3 Enhancement Delta (Baseline → Enhanced)

Metric	Baseline	Enhanced	Delta	Improved?
Groundedness	3.00	2.95	−0.05	No
Relevance	3.10	3.40	+0.30	Yes
Context Precision	2.15	2.25	+0.10	Yes
Citation Precision	0.99	0.98	−0.01	No
Source Recall	0.37	0.40	+0.03	Yes

The query rewriting/decomposition enhancement produced modest gains in relevance (+0.30), context precision (+0.10), and source recall (+0.03), but did not improve groundedness (−0.05) or citation precision (−0.01). The biggest improvement came in synthesis queries, where relevance jumped from 2.0 to 3.5, indicating that sub-query decomposition helped retrieve more diverse and relevant chunks for cross-source comparisons. However, the core bottleneck, low source recall across the board, persists, suggesting the embedding model struggles to match queries to the specific expected source documents.

5. Representative Failure Cases

Five representative failures were identified across both pipelines. They cluster around two root causes:

Failure 1 : Retrieval Miss (D02, D04, D07): The most frequent failure mode is the retrieval of irrelevant source chunks, leaving the expected source entirely absent from the top-K results. For query D02 (BERT CO2 emissions per Strubell et al.), the system retrieved chunks from dodge2022 and lannelongue2021 instead of strubell2019, causing the LLM to hallucinate specific statistics (754,407 gCO2e) from parametric memory. Similarly, D04 (Patterson et al. factors for reducing emissions) retrieved only henderson2020 and faiz2024 chunks, making it impossible to answer the question. D07 (carbon tracking tools) retrieved lacoste2019 chunks but the LLM fabricated details about a tool URL and features not present in the chunks. These failures all stem from the embedding model failing to semantically match author-specific or method-specific queries to the correct documents.

Failure 2 : Parametric Hallucination (S02 baseline, D07 baseline): When the retriever fails to surface relevant evidence, the LLM compensates by drawing on its training knowledge instead of strictly grounding in the provided chunks. In S02 (Luccioni vs. Patterson assumptions), no Luccioni et al. chunks were retrieved, yet the model attempted to frame a comparison. In D07, the model fabricated a specific URL for the ML Emissions Calculator that was not present in any retrieved chunk. The generation prompt instructs the model to cite only from provided context, but it does not consistently refuse to answer when context is insufficient for direct queries.

Failure 3 : Low Relevance on Edge Cases (E01, E04): Edge-case queries that expect the system to say "no evidence found" sometimes score poorly on relevance (1/4) even though they correctly flag missing evidence and score 4/4 on groundedness. This reflects a tension in the evaluation rubric: the judge penalizes an answer that does not "address the question" even when the correct answer is that the corpus lacks evidence. This is a scoring artifact rather than a true system failure, but it inflates the appearance of poor performance on edge cases.

6. Interpretation and Next Steps

Key findings: The system's strongest capability is citation precision (≥0.97 across all configurations), meaning that when the system cites a source, the citation almost always resolves to real text. Its weakest capability is source recall (0.37–0.40), indicating that the retriever frequently fails to surface the expected documents. Groundedness hovers at the warn threshold (~3.0), primarily because retrieval misses force the LLM to either refuse to answer or hallucinate from parametric knowledge. The query rewriting enhancement helped most with synthesis-type queries but did not resolve the fundamental retrieval gap.

Recommended improvements for Phase 3: (1) Hybrid retrieval combining BM25 keyword search with vector search to improve recall on author-name and method-specific queries. (2) A cross-encoder reranker to improve context precision after initial retrieval. (3) Stronger generation guardrails that force the model to explicitly refuse answering when retrieved chunks do not contain evidence from the queried source, rather than falling back on parametric knowledge. (4) Adjustment of the LLM-as-Judge rubric for edge cases so that correctly identifying missing evidence is scored as relevant.

7. Reproducibility

All dependencies are pinned in requirements.txt. The full evaluation pipeline is reproducible via four commands:

```
pip install -r requirements.txt
python -m src.ingest.download_sources
python -m src.ingest.ingest
python -m src.eval.evaluation --mode both
python -m src.eval.generate_report
```

The FAISS index and chunk store are fully reproducible from scratch using the data manifest. Run logs for all 40 evaluated queries (20 baseline + 20 enhanced) are stored as machine-readable JSON.