A Graph-Retrieval-Augmented Generation Framework Enhances Decision-Making in the Circular Economy

Yang Zhao[1], Chengxiao Dai[2], Dusit Niyato[3], Chuan Fu Tan[1], Keyi Xiang[3], Yueyang Wang[4], Zhiquan Yeo[1], Daren Tan Zong Loong[1], Jonathan Low Zhaozhi[1], Eugene H.Z. HO[1]

[1]Singapore Institute of Manufacturing Technology (SIMTech), Agency for Science, Technology and Research (A*STAR), 2 Fusionopolis Way, Innovis #08-04, 138634, Republic of Singapore.

[2]School of Computer Science, Faculty of Engineering, The University of Sydney, Building J12, 1 Cleveland Street, Darlington, NSW, 2008, Australia.

[3]College of Computing and Data Science, Nanyang Technological University, 50 Nanyang Ave, 639798, Republic of Singapore.

[4]Faculty of Science, National University of Singapore, 6 Science Drive 2, 117546, Republic of Singapore.

Contributing authors: Zhao_Yang@simtech.a-star.edu.sg; daicxx1226@gmail.com; DNIYATO@ntu.edu.sg; Tan_Chuan_Fu@simtech.a-star.edu.sg; KEYI003@e.ntu.edu.sg; e0726339@u.nus.edu; zqyeo@simtech.a-star.edu.sg; tanzld@simtech.a-star.edu.sg; Jonathan_Low@simtech.a-star.edu.sg; eugene_ho@simtech.a-star.edu.sg;

Abstract

Large language models (LLMs) hold promise for sustainable manufacturing, but often hallucinate industrial codes and emission factors, undermining regulatory and investment decisions. We introduce CircuGraphRAG, a retrieval-augmented generation (RAG) framework that grounds LLMs outputs in a domain-specific knowledge graph for the circular economy. This graph connects 117,380 industrial and waste entities with classification codes and GWP100 emission data, enabling structured multi-hop reasoning. Natural language queries are translated into SPARQL and verified subgraphs are retrieved to ensure accuracy and traceability. Compared with Standalone LLMs and Naive RAG, CircuGraphRAG achieves superior performance in single-hop and multi-hop question answering, with ROUGE-L F1 scores up to 1.0, while baseline scores below 0.08. It also improves efficiency, halving the response time and reducing token usage by 16% in representative tasks. CircuGraphRAG provides fact-checked, regulatory-ready support for circular economy planning, advancing reliable, low-carbon resource decision making.

Keywords: Circular Economy, Large Language Model, RAG, GraphRAG.

1 Introduction

The global transition to a circular economy promises substantial improvements in resource efficiency and waste reduction, whereby the by-products of one industrial process can serve as input to another, thus reducing waste and conserving resources [1, 2]. Despite its considerable potential, many organizations face persistent barriers, including fragmentation of data, limited trust, and stringent regulatory requirements, which hinder their ability to align resource availability with sustainability objectives [3–5]. In addition, material flow, energy use, and emission data are frequently located in isolated systems, undermining collective efforts to minimize waste [6, 7]. Even when potential opportunities arise, the lack of standardized terminology, divergent measurement protocols, and intellectual property concerns often restrict open data sharing, thus impeding the widespread adoption of circular practices [8, 9].

Currently, environmental, social, and governance (ESG) reporting frameworks [10], such as the Global Reporting Initiative (GRI) [11], the Sustainability Accounting Standards Board (SASB) [12], and the Climate-related Financial Disclosure Task Force (TCFD) [13], have gained prominence in guiding corporate transparency and investment decisions. These frameworks require auditable disclosures on waste streams, resource utilization, and environmental impacts, aligning closely with the objectives of a circular economy. However, bridging data fragmentation, establishing trust across organizations, and meeting compliance demands remain non-trivial tasks.

Standardized databases and systematic monitoring frameworks have emerged to address these data-related challenges [14, 15]. Classification systems such as the European Waste Catalog (EWC) [16] and the Nomenclature of Economic Activities (NACE) [17] provide universal taxonomies for categorizing resources and waste, thus reducing intersectoral misalignment. These foundational structures are suitable for the construction of knowledge graphs [18], where entities such as waste streams, production facilities are represented as nodes, and their relationships, including “can be processed by,” and “located in” are encoded as semantic links. Typically, these links are implemented using the Resource Description Framework (RDF) [19, 20], ensuring structured and machine-readable data across multiple domains.

Recent advances in large language models (LLMs) offer expert-level insight and decision support in various fields, including healthcare and environmental management [21]. However, LLMs can produce inaccurate or fabricated content due to hallucination [22–24], reinforcing the need for robust and vetted databases. Retrieval-augmented generation (RAG) [25, 26] partially addresses this challenge by integrating external knowledge into the inference process. An advanced variant, GraphRAG [27], exploits the interconnected nature of knowledge graphs to perform multi-hop reasoning, producing more reliable outputs than traditional RAG pipelines based solely on vector similarity [28, 29]. By systematically linking nodes through explicit relationships, GraphRAG retrieves contextually relevant knowledge, thus improving precision and mitigating tangential data [30]. In addition, this graph-based structure clarifies the chain of reasoning, allowing users to trace how each data point contributes to the final output, mitigating hallucinations, and increasing user trust.

To align with climate mitigation targets, it is also critical to integrate emission factor data. The 100-year Global Warming Potential (GWP100) indicator provides a standardized benchmark to assess the climate impacts of industrial processes [31, 32]. Reflecting GWP100 in transactions, stakeholders can rank or filter synergy opportunities based on greenhouse gas footprints, thus making decisions that comply with ESG standards and advance sustainability goals.

Together, these frameworks and technologies create the foundation for a robust, data-driven platform that can effectively enable circular economy synergies. Despite these promising developments, direct applications of knowledge graphs and GraphRAG within the circular economy remain limited. Existing standards and frameworks provide a foundation for modeling waste and resource data but lack integration into a cohesive platform capable of supporting large-scale synergy discovery and continuous sustainability assessments [33, 34].

In this paper, we detail how CircuGraphRAG synergistically integrates knowledge graph technology, RAG, and ESG metrics to advance data-driven circular economy practices. Specifically, we build a harmonized industrial-waste knowledge graph that links industries, waste classifications, materials, and environmental factors (GWP100) across multiple standards (ISIC, NACE, EWC), enabling structural and semantic exploration. We then present a knowledge graph enabled RAG architecture that incorporates ontology-aligned industrial knowledge and environmental metrics, supporting structured retrieval and multi-hop reasoning for low-carbon industrial symbiosis. Our approach further employs a light-weight library of 18 SPARQL query templates, automatically selected and parameterized by the LLMs, to facilitate both single-hop lookups and multi-hop synergy discovery without handwritten rules. In addition, a hybrid ranking mechanism uses GWP100 to order query results, guiding practitioners toward low-carbon waste-to-resource pathways. We also contribute a reproducible graph reasoning benchmark for code alignment, numeric accuracy, and industrial symbiosis scenarios. Finally, we demonstrate model-agnostic empirical gains across various LLMs, showing that CircuGraphRAG consistently reduces hallucinations and accelerates inference compared to Naive RAG, without diminishing answer completeness.

The abbreviations used in the paper are summarized in the Appendix 6.9.

2 Results

Developing robust waste-to-resource strategies is crucial to minimize environmental impacts and advance circular economy goals. This section presents an extensive evaluation of CircuGraphRAG across realistic waste-to-resource management queries, illustrating how knowledge-graph-driven retrieval substantially enhances the discovery of reuse pathways. We compare our CircuGraphRAG against several baselines on three primary tasks: structured data extraction (single-hop queries) and multi-hop reasoning including synergy identification. Quantitative metrics are complemented by an analysis of computational overhead and template matching ablation study, as well as an assessment of the multi-round consistency of LLMs reasoning.

2.1 Quantitative Results for Single-Hop Queries

Single-hop queries consist of directly retrieving a single fact or entity from the knowledge graph. A representative query is *"Which resource corresponds to EWC code 080121 and HS code 810330?"* in Appendix 6.1. Then, Appendix 6.2 summarizes the results on three distinct single-hop queries (Cases 1–3).

2.1.1 Overall Performance

CircuGraphRAG consistently outperforms both Standalone LLM and Naive RAG across all examined LLMs (Llama, Qwen, DeepSeek) in terms of ROUGE-L F1 scores. These improvements are visualized in Figure 1a, which illustrates the performance differences across models and test cases in three dimensions. Detailed numerical results can be found in Appendix 6.2, highlighting the significant gains enabled by domain-constrained retrieval.

2.1.2 Precision and Recall Gains

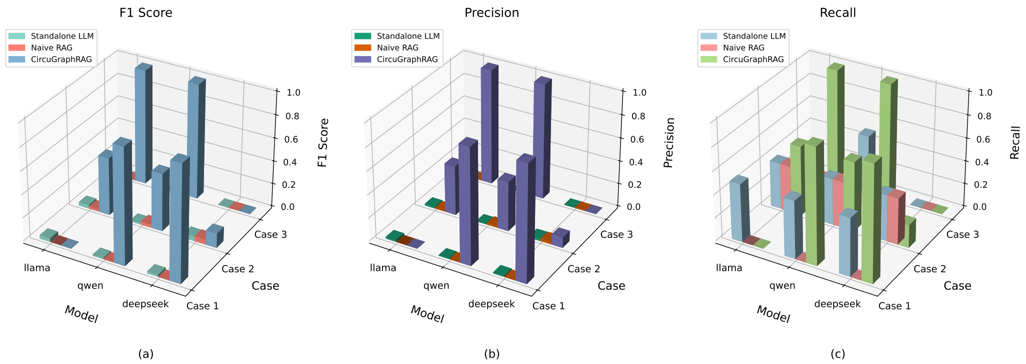
In Case 1, CircuGraphRAG with Qwen and with DeepSeek both achieve a perfect ROUGE-L recall (1.0). In contrast, Standalone LLM approaches frequently produce incomplete or inaccurate responses, suggesting hallucinations or a lack of contextual grounding. However, CircuGraphRAG anchors the query to specific domain edges, such as `iskg:hasHSCode` and `iskg:hasEwcCode`, to reliably match the correct entity.

In Case 2, CircuGraphRAG improves ROUGE-L precision and recall over baselines. For example, CircuGraphRAG with Qwen achieves a ROUGE-L precision of 0.4286, a substantial gain over Naive RAG with Qwen (0.0150). These improvements indicate that forcing query resolution through valid code and entity relationships (e.g., `iskg:hasNaceCode`) filters out spurious completions. From a regulatory point of view, an incorrect assignment of provider or resource codes (for example, NACE 3821 vs. 3822) could lead to non-compliance.

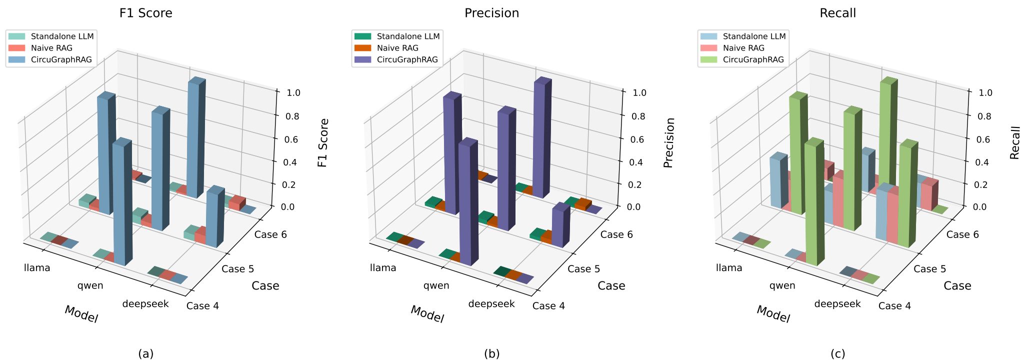
2.1.3 Numeric Retrieval

Numerical accuracy is critical for compliance driven domains, particularly for GWP100. In Case 3, CircuGraphRAG with both Qwen and Llama correctly retrieves the minimal GWP100 value (0.008930631), achieving an ROUGE-L F1 score of

1.0. All baseline approaches either return a spurious figure or no result, underscoring the difficulty of numeric retrieval within unconstrained LLMs. Since small deviations in emission metrics can misrepresent environmental impact assessments, CircuGraphRAG's exact retrieval of numeric data is vital for reliable industrial decision-making.



(a) Single-hop QA: ROUGE-L F1, precision, and recall across Cases 1–3.



(b) Multi-hop QA: ROUGE-L F1, precision, and recall across Cases 4–6.

Fig. 1: 3D performance comparison of different LLMs and retrieval methods on single-hop QA tasks.

2.2 Quantitative Results for Multi-Hop Reasoning and Synergy Identification

Multi-hop queries assess a model's capacity to traverse multiple linked relationships, an essential capability for identifying synergies in industrial symbiosis. For instance, verifying that a receiver of one resource can act as a provider of another often requires navigation through various classification codes (e.g., EWC, HS, NACE, CPA) and properties (e.g., `iskg:hasGwp100`). As shown in Figure 1b, CircuGraphRAG consistently outperforms both Standalone LLM and Naive RAG in Cases 4–6, achieving notably higher ROUGE-L F1 scores.

In Case 4, the query determines which CPA code and category apply to the resources generated under a specific NACE code. Both Standalone LLM and Naive RAG yield a ROUGE-L F1 of 0.0 across all language models. CircuGraphRAG with Llama or DeepSeek also returns 0.0, yet CircuGraphRAG with Qwen achieves 1.0, indicating that a sufficiently capable model can take advantage of the constraints based on the knowledge graph.

Moreover, Case 5 extends this concept by examining whether receivers under a CPA code can later serve as providers of a second resource (e.g. "waste polystyrene"). CircuGraphRAG with Llama or Qwen achieves a perfect ROUGE-L F1 of 1.0, whereas Naive RAG and Standalone LLM each fall below 0.08 in most configurations, underscoring that purely generative reasoning is prone to errors without explicit domain guidance. CircuGraphRAG with DeepSeek performs moderately well (ROUGE-L F1=0.4615), though not matching the near-ideal scores of CircuGraphRAG with Qwen or Llama.

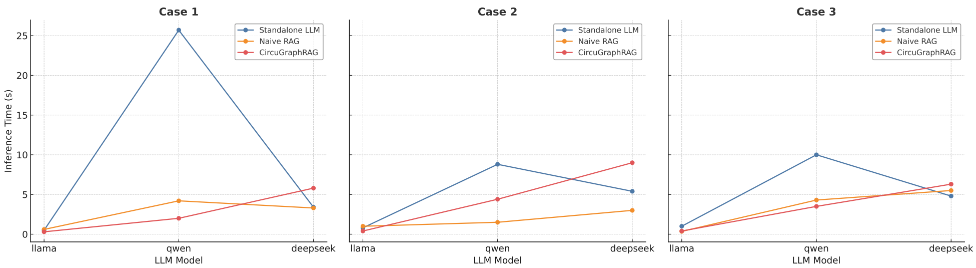
In Case 6, the query identifies providers of a resource coded EWC 070213 that can also receive "aluminum scrap." CircuGraphRAG with Qwen again achieves a ROUGE-L F1 of 1.0, while CircuGraphRAG with Llama and DeepSeek remain at 0.0, and the baseline approaches exhibit limited precision below 0.035. These results demonstrate that CircuGraphRAG significantly outperforms baseline methods in complex multi-hop queries, particularly when structural constraints are applied. However, performance differences across LLMs remain, and some models still require strategy calibration or further fine-tuning to fully leverage knowledge graph-enhanced reasoning.

From a circular economy perspective, this level of multi-hop precision is crucial for industrial symbiosis, in which by-products from one facility become feedstock for another. By encoding explicit relationships in a knowledge graph, CircuGraphRAG reduces reliance on heuristic inference, thus mitigating misclassification risks and facilitating data-driven synergy discovery. Overall, these results confirm that CircuGraphRAG enhances multi-hop reasoning, particularly when deployed with capable LLMs, thus improving the identification of multi-step resource flows and advancing circular economy's objectives. Supporting metrics for this evaluation are presented in Appendix 6.3.

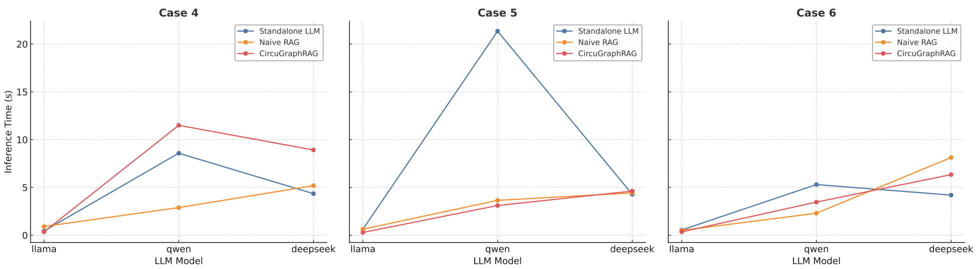
2.3 Computation Cost and Token Usage

We further evaluated CircuGraphRAG in terms of token consumption and inference time across various LLM back-ends and task types. Although the framework incorporates additional retrieval components including template matching (TM), query merging (QM), and querying with recovered context (QRC), which might increase computational overhead, our results indicate that CircuGraphRAG consistently reduces both total token usage and inference latency. These gains are attributed to the construction of more compact, domain-specific input contexts that constrain the language model's generative scope and streamline the reasoning process.

In single-hop queries (Figure 2a), CircuGraphRAG with Qwen processes fewer tokens more quickly than Naive RAG with Qwen or Standalone Qwen, as evidenced in Case 1 (2,009 tokens vs. 2,381 tokens and over 10,000 tokens) while doubling inference speed. The approach may incur higher token usage in more complex tasks (Case 2) due to broader subgraph lookups, but yields more reliable answers by leveraging domain-specific evidence. In contrast, Standalone Qwen tends to generate excessively long outputs with correspondingly higher latency.



(a) Single-hop cases (Cases 1–3).



(b) Multi-hop cases (Cases 4–6).

Fig. 2: Inference time comparison across different models and methods for (a) single-hop and (b) multi-hop QA tasks.

Multi-hop queries often exacerbate off-topic content unless retrieval is carefully guided. In Case 5 (Figure 2b), Standalone Qwen outputs 8,677 tokens in 21.35 s, while CircuGraphRAG with Qwen trims responses to 2,733 tokens only in 3.11 s. Although certain queries (e.g., Case 4) require more extensive retrieval (6,041 tokens over 11.50 s), CircuGraphRAG is the only method that achieves an F1 score of 1.0 for this scenario. Across all LLMs, CircuGraphRAG constrains generative pathways by embedding domain relationships, balancing overhead with enhanced multi-hop accuracy.

Overall, these observations show that CircuGraphRAG effectively reduces superfluous text generation while sustaining high factual fidelity. When additional tokens are needed, the resulting improvements in answer quality and reliability offset the higher computational cost, demonstrating the potential of a domain-targeted retrieval structure for large-scale industrial knowledge queries.A detailed comparison of token usage and inference latency across models and methods supporting these findings is presented in Appendix 6.4 and Appendix 6.5.

2.4 Template Matching Ablation Study

Since structural templates play a central role in shaping how the model identifies and composes query paths, we conduct an ablation study to investigate how different levels of template guidance affect the reasoning process in CircuGraphRAG. Specifically, our objective is to evaluate whether removing or weakening the use of templates impacts the accuracy of downstream SPARQL query generation and final answers.

We consider three template configurations: (1) **with template**, where the system utilizes explicitly defined query structures to guide the reasoning process; (2) **no template**, where the LLM must generate the SPARQL query path without structural guidance; and (3) **fuzzy template**, where only partial structural hints (e.g. entity types or field names) are provided without complete logical templates. For each setting, we evaluate the accuracy of the answer in three LLMs under the same task input. To isolate the influence of template guidance, we conducted this study in a representative single-hop QA case where the expected output depends on precise entity-code mapping.

Table 1: Accuracy Comparison under Different Template Matching Conditions. (A ✓ indicates a correct answer; a ✗ indicates an incorrect one.)

Template Type	llama	qwen	deepseek
With Template	✓	✓	✓
No Template	✗	✗	✗
Fuzzy Template	✓	✓	✗

As shown in Table 1, the performance of different models varies significantly across template configurations, further validating the critical role of predefined templates in the CircuGraphRAG system. Under the **with template** condition, all three models achieve 100% query matching accuracy. This demonstrates that explicitly injected templates not only provide structural guidance, but also normalize the reasoning path and entity operation process, directly determining the quality and stability of SPARQL query generation.

In contrast, when the templates are completely removed (**no template**), none of the models can generate correct SPARQL queries, and the accuracy drops to 0. This result clearly demonstrates that even in single-hop question answering scenarios, large language models struggle to reconstruct the necessary triple patterns and query logic without explicit structural templates. The absence of templates leads to significant errors in entity selection, property targeting, and predicate direction, often resulting in failed path construction or invalid answers.

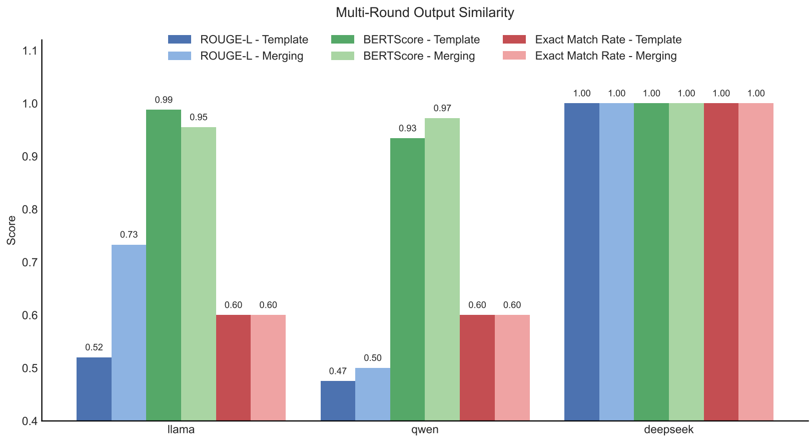
Under the **fuzzy template** condition, the system provides only partial structural cues. Llama and Qwen still maintain relatively high accuracy, whereas DeepSeek shows a notable performance drop. This suggests that while fuzzy template hints can assist in constructing basic reasoning structures. However, without complete template logic, query path generation becomes unstable and the final answers less reliable, particularly for models with limited structural understanding.

2.5 Multi-Round Consistency of LLMs Reasoning

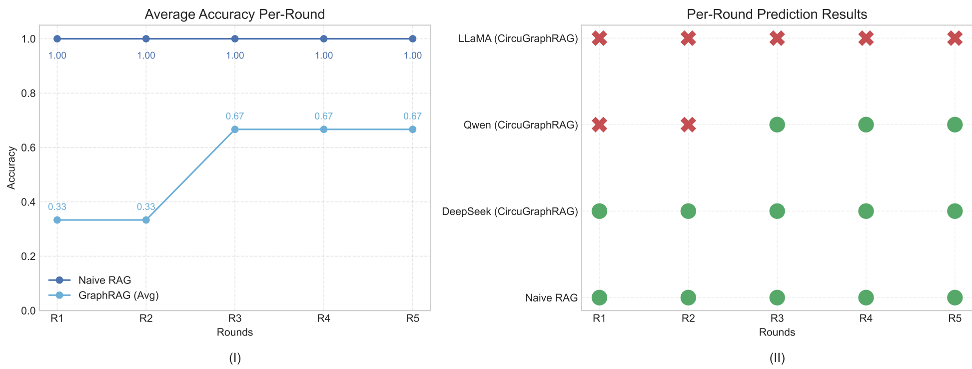
To evaluate the robustness of the structured reasoning framework against the inherent uncertainty in LLM output, we designed a set of experiments to examine whether variability during the reasoning phase of LLM affects the overall performance of CircuGraphRAG. Although the system leverages template matching and SPARQL query construction to enable structured question answering supported by a knowledge graph, the generation of these structures is still fundamentally dependent on the reasoning capability of the LLM. We hypothesize that if the LLM exhibits instability in template recognition or query path merging, it may lead to downstream retrieval failures or incorrect paths, ultimately degrading the answer accuracy.

We designed two sub-experiments, and to ensure experimental controllability, we conducted the evaluation on a representative single-hop question answering case. The first part evaluates the structural output consistency of the LLM in two key stages: *template matching* and *query merging*. Specifically, we compute three metrics over five repeated runs: (1) **ROUGE-L** for lexical similarity, (2) **BERTScore** for semantic similarity, and (3) **Exact Match Rate** to measure whether the generated structures remain identical across runs. The second part examines whether structural generation differences affect the final answer quality. We record the answer accuracy of CircuGraphRAG, Naive RAG, and Standalone LLM over five runs for each input and compare their accuracy levels.

According to Figure 3, the results confirm that LLMs exhibit significant instability in two critical stages of structured generation, template matching and query merging. Although DeepSeek maintains complete consistency across multiple reasoning rounds, both Llama and Qwen achieve only 60% exact match rates in both stages, indicating substantial variability in symbolic output structures (as reflected in lower ROUGE-L scores). Although the semantic content remains largely consistent (as shown by high



(a) Multi-round output similarity across LLMs and stages.



(b) I. Average accuracy of CircuGraphRAG vs. Naive RAG over 5 rounds. II. Per-round correctness indicators for each model.

Fig. 3: Multi-round consistency and accuracy evaluation of CircuGraphRAG.

BERTScores), such structural fluctuations can still lead to erroneous retrieval paths, ultimately compromising the accuracy of the final answers.

In contrast, non-structured RAG methods such as Naive RAG and Standalone LLM demonstrate greater tolerance to reasoning variance, as they are not bound by rigid structural dependencies. This highlights a key trade-off: while structured QA frameworks enhance interpretability and control, they also amplify the adverse impact of LLM reasoning instability.

2.6 Error Analysis and Qualitative Observations

While CircuGraphRAG substantially reduces hallucinations and numeric errors, two primary limitations remain:

- **Knowledge Gaps:** Some specialized or newly introduced waste streams may be missing in the knowledge graph. In such cases, CircuGraphRAG returns partial or incomplete results.
- **Vague User Prompts:** Overly broad queries can mismatch the available templates, producing off-target or partial retrieval. By contrast, standalone LLMs and naive RAG often produce confident but incorrect code assignments or invented numerical values.

These limitations underscore the importance of regularly updating the ontology and refining domain-specific query templates to ensure coverage of emerging regulations and precise user intent.

2.7 Summary of Findings

Our experiments demonstrate that aligning an LLM with a structured knowledge graph offers significant accuracy and efficiency gains for waste-to-resource management. CircuGraphRAG consistently delivers higher ROUGE-L scores in single and multihop tasks, reduces hallucinations and numeric inaccuracies, and in many queries lowers the overall token footprint by 15–16% compared to naive retrieval enhancement. Its strong synergistic detection performance and precise code alignments highlight the potential of the system to navigate complex industrial ecology regulations and accelerate the real-world adoption of circular economy practices.

3 Discussion

Our findings illustrate that linking an LLM to a structured knowledge graph significantly improves accuracy, numeric consistency, and code alignment in waste-to-resource management. The examples presented in Appendix 6.7 and Appendix 6.8 highlight how CircuGraphRAG outperforms both standalone LLMs and naive retrieval-augmented systems in representative queries, ranging from simple lookups from "identifying the correct category for polyurethane waste" to multi-hop reasoning such as "matching NACE code 3821 to CPA code 382150".

3.1 Key Advances

3.1.1 Reduced Hallucinations in Single-Hop Queries

We selected the classification of "waste polyurethane" as a representative single-hop query to evaluate the reasoning performance of CircuGraphRAG. While the ground truth specifies "construction and demolition wastes," Standalone LLMs frequently produced overgeneralized or inconsistent categories, such as "municipal solid waste." In contrast, CircuGraphRAG accurately identified the correct classification through its graph-based retrieval mechanism, which links "waste polyurethane" to its corresponding relationship (e.g., `hasCategory`) in the knowledge graph. In real-world applications, classification codes are often governed by regulatory standards for safe disposal or recycling, making the reduction of such hallucinations essential. Detailed results for this case are presented in Appendix 6.7.

3.1.2 Enhanced Multi-Hop Reasoning for the Identification of Synergies

Multi-hop queries often involve combining multiple relationships, such as mapping a provider's NACE code (e.g., 3821) to the CPA code corresponding to its generated resources (e.g., 382150). CircuGraphRAG achieves exact matches in such scenarios, where both Standalone LLMs and naive retrieval methods typically fail, underscoring the system's capacity to *verify each step* against structured ontology links. This capability to handle multi-hop reasoning is essential for identifying opportunities for industrial symbiosis, such as matching metal slags from steelmaking with cement production, thereby supporting circular-economy initiatives. Illustrative multi-hop examples can be found in Appendix 6.8.

3.1.3 Traceable Explanations and Regulatory Relevance

Beyond numeric metrics, CircuGraphRAG ensures that each retrieval step is transparent, as illustrated in the "Query Merging" blocks in Appendix 6.7 and Appendix 6.8. Domain experts and regulators can audit exactly how the system arrived at a particular classification code or emission factor. Given the potential legal and environmental ramifications of an incorrect or fabricated classification (e.g., labeling a hazardous waste as non-hazardous), this built-in transparency meets a critical need in regulated industries.

3.2 Practical Implications

The improved accuracy and explainability of the system address key challenges in industrial ecology.

3.2.1 Faster and More Reliable Synergy Discovery

Appendix 6.7 and Appendix 6.8 feature typical queries in which organizations seek to identify compatible receivers for by-products. CircuGraphRAG reduces the manual overhead of cross-referencing waste codes and emission thresholds, thus surfacing reuse pathways that a purely text-based approach might overlook.

3.2.2 Minimizing Incorrect Classifications

Even minor classification errors (e.g., mislabeling polyurethane as municipal waste) can lead to non-compliance or suboptimal resource routing. The examples show that CircuGraphRAG's graph-anchored retrieval routinely supplies the correct code alignment, enabling more rigorous decision-making for facility-level or regional-scale waste management.

By systematically linking classification codes, GWP100, and facility attributes, CircuGraphRAG accelerates large-scale circular initiatives. Its subgraph retrieval also provides an auditable trail that reassures regulators and other stakeholders, who can confirm that each proposal meets legal or emissions requirements before implementation.

4 Methods

We propose CircuGraphRAG, a retrieval-augmented LLM framework that integrates multiple industrial classification systems, environmental impact factors, and textual embeddings into a unified knowledge graph. By automating single-hop and multi-hop queries, CircuGraphRAG identifies feasible "waste-to-resource" routes across industries, highlighting those with low GWP100 emissions. Our approach tackles two core challenges of sustainable industrial ecology:

- the diversity and complexity of classification schemes for waste and industrial processes, and
- the need to ground LLM responses in reliable, structured data to avoid hallucinations.

The following sections describe each methodological step, from dataset construction to the final synergy recommendations ranked by GWP100.

4.1 Dataset Overview

Our experiments make use of 3,896 entries from the "Waste Treatment and Recycling" sector of the Ecoinvent dataset [35], each mapped to International Standard Industrial Classification of All Economic Activities (ISIC), Central Product Classification (CPC), Nomenclature statistique des activités économiques dans la Communauté européenne (NACE), Wirtschaftszweige (WZ), Classification of Products by Activity (CPA), European Waste Catalogue (EWC), Harmonized System (HS), and Singapore Standard Industrial Classification (SSIC) codes, as well as the respective impact factor GWP100. We merge domain-specific information such as facility locations, emissions data, provider and receiver identities, and regulatory constraints into a unified knowledge graph. This preserves the original code mappings and introduces semantic links across datasets. In practice, we harmonize codes from multiple classification systems to ensure cross-regional compatibility. The resulting knowledge graph includes the following:

- **117,380 nodes**, each representing a discrete resource entity (e.g., waste streams, by-products, industrial facilities, emission flows and regulatory guidelines).
- **753,145 edges**, capturing relationships such as `hasGwp100` (mapping a resource to its *GWP100*), `hasProvider`, `hasReceiver` (linking facilities that generate or accept a given waste stream), and `hasResource` (connecting a by-product to relevant industrial processes).
- An **integrated ontology** that consolidates EWC, NACE, ISIC, and SSIC coding, normalizing terminology that would otherwise vary across geographies or regulatory bodies.

4.2 Dataset Construction

We draw on the Ecoinvent database [35], a comprehensive repository of industrial processes, resource flows, and emissions data as shown in Figure 4. Each record is enriched with standard classification codes, such as ISIC, NACE, SSIC, WZ, EWC, HS, and CPA, thus enabling both *industry-oriented* and *waste-oriented* analyses:

- **Industry-oriented queries:** Identify synergies between industries based on shared or compatible classification codes, as shown in Figure 4a.
- **Waste-oriented queries:** Pinpoint potential receivers for a given waste stream aligned with regulatory codes and environmental metrics, as shown in Figure 4b.

Where direct crosswalks are incomplete, we apply sentence-embedding techniques (`multi-qa-mpnet-base-cos-v1`) to infer plausible mappings from textual descriptions. Newly inferred codes are flagged for domain experts to validate. This harmonized dataset provides a robust basis for constructing our knowledge graph, ensuring consistent references for industrial activities and waste categories across multiple sectors and jurisdictions.

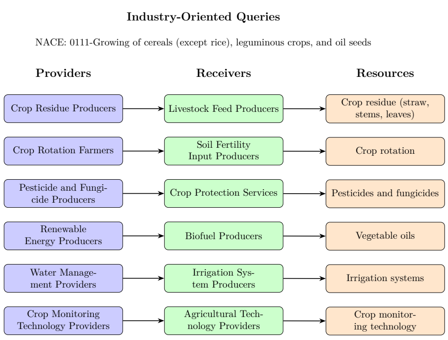
4.3 Ground Truth Dataset Construction

To rigorously measure the accuracy of the response, we developed a set of canonical queries and the corresponding *reference ground truth*. Table 2 illustrates the variety of single-hop and multi-hop queries used to evaluate the systems. Each query–answer pair underwent a multi-hop curation process to ensure industrial relevance and technical correctness:

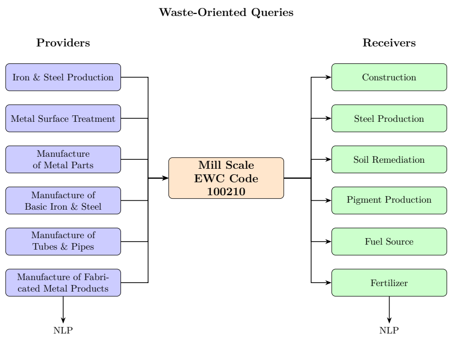
1. **Query Selection:** We identified representative questions reflecting real-world needs in waste-to-resource management, including code alignment (EWC, HS, NACE, CPA) and key attributes (GWP100). These queries span single-hop lookups (e.g., retrieving a code-specific entity) and multi-hop reasoning (e.g., applying regulatory constraints across multiple entities).
2. **Domain Annotation:** We cross-referenced official coding guidelines and our expanded knowledge graph to establish gold-standard answers. Verification was centered on correct waste code matching, valid provider–receiver relationships, and consistent metadata.
3. **Cross-Checking:** Each annotated answer was tested against our knowledge graph to confirm that the declared relationships were unambiguously retrievable. Any inconsistencies resulted in further refinements to the knowledge graph or the ground truth.

Subsequently, we evaluated the performance of our proposed CircuGraphRAG approach on two primary tasks: answering *single-hop* and *multi-hop* queries. These tasks range from straightforward fact retrieval, such as obtaining a GWP100 value, to multi-hop reasoning that involves identifying by-products and matching them with suitable receivers under specific regulatory constraints. We compare CircuGraphRAG with two baselines:

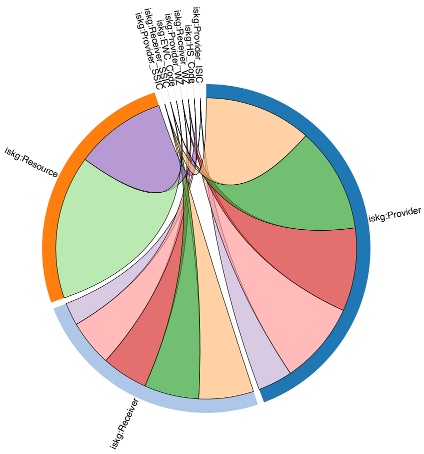
1. **Standalone LLM:** An LLM in isolation (Llama, Qwen, DeepSeek) without explicit retrieval.
2. **Naive RAG:** A RAG approach that lacks fine-grained graph constraints.



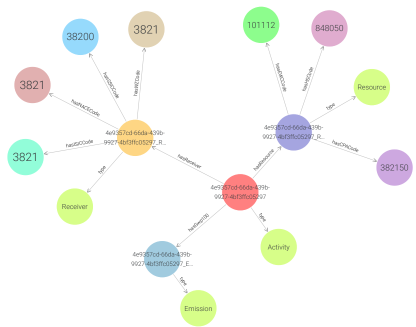
(a) Industry-Oriented Queries (NACE: 0111).



(b) Waste-Oriented Queries (EWC: 100210).



(c) Class Relationship.



(d) Visual Graph of Waste: 4e9357cd-66da-439b-9927-4bf3ffc05297.

Fig. 4: An overview of the structure of our knowledge graph database.

4.4 Knowledge Graph Construction

1. **Providers:** Entities or processes that generate secondary materials or by-products.
2. **Receivers:** Facilities that reuse or recycle these materials.
3. **Resources:** The actual material flows, annotated with attributes such as GWP100 or hazardous status.

Properties (e.g., `iskg:hasEWCCode`, `iskg:hasGwp100`) link these classes as shown in Figure 4c, enabling SPARQL queries for fine-grained investigations of waste-to-resource pathways. Using canonical URIs for classification codes, we maintain semantic clarity even when bridging multiple standards (ISIC, NACE, EWC, etc.) as shown in Figure 4d.

4.5 The CircuGraphRAG Framework

CircuGraphRAG unifies knowledge graph queries with large language models in a retrieval-augmented generation pipeline (see Figure 5). Its main components include :

4.5.1 Text-Based Query

Users submit queries in free text form, often specifying a particular industrial code or waste type (e.g. “Find the provider with NACE code 3821 producing waste cement"). An LLM then parses this prompt to extract key concepts (e.g., “NACE 3821,” “waste cement”).

4.5.2 Entity Classification & Vector Indexing

Each user query is parsed to determine whether it targets a *provider* or a *receiver*. Potential entities are encoded into 384-dimensional embeddings using `all-MiniLM-L6-v2` and stored in two Facebook AI Similarity Search (FAISS) indexes (one for providers, one for receivers). For each query embedding $\mathbf{q} \in \mathbb{R}^d$, we retrieve the top-$k$ most similar entity vectors $\{\mathbf{e}_1, \ldots, \mathbf{e}_k\}$ by maximizing the inner product

$$\mathrm{score}(\mathbf{q}, \mathbf{e}_i) = \langle \mathbf{q}, \mathbf{e}_i \rangle = \sum_{j=1}^{d} q_j \cdot e_{i,j}.$$

This is implemented efficiently using `faiss.IndexFlatIP`, which ranks all entity vectors in descending order of similarity. Top-k candidates are retrieved for downstream reasoning and reranking.

4.5.3 Knowledge Graph-based Retrieval and Reasoning

Based on Algorithm 1, the knowledge graph-based retrieval process uses an LLM-driven template matcher to interpret user queries and produce appropriate SPARQL statements. We maintain a TemplateLibrary of 18 distinct SPARQL templates, each designed to address common query intentions (for example, “Find all receivers for a specific waste stream,” “Locate facilities by classification code,” etc.). In Step 1, the LLM identifies relevant entities (for example, a facility code or the name of a resource) and selects all templates that match the semantic intent of the user’s question. These matching templates, often more than one, have placeholders (such as `?providerCode` or `?resourceName`) that are filled with information extracted from the query.

In Step 2, if multiple templates are selected, the LLM determines how to merge them according to the query requirements. Depending on whether the user needs

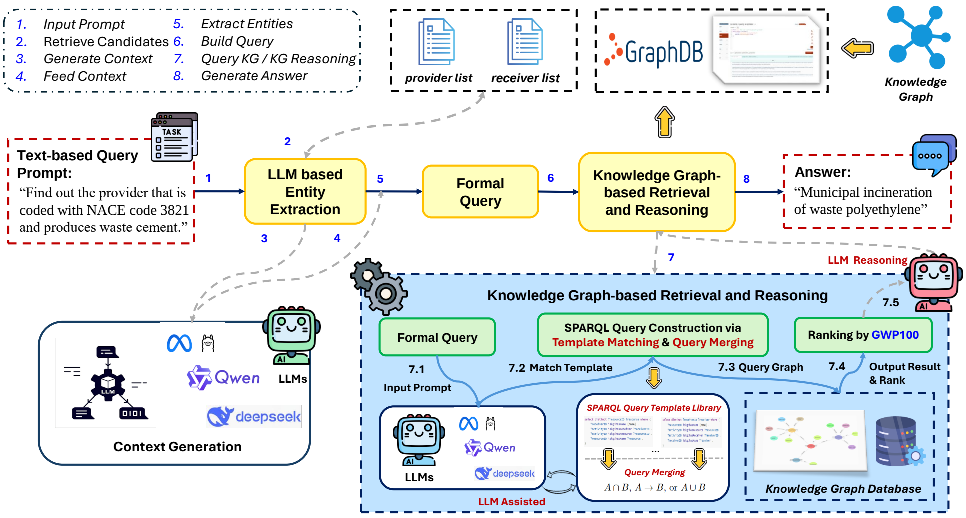


Fig. 5: The CircuGraphRAG Pipeline.

intersecting results, chained filtering, or broader coverage, the templates are combined using operations, including intersection, chaining, or union. This automated merging ensures that all relevant aspects of the user's request are handled in a single FinalQuery.

Subsequently, in Step 3, we execute the FinalQuery against the knowledge graph and retrieve a structured result set that contains facts such as the matching of facility names, classification codes, and values GWP100. We then rank the results by ascending GWP100, ensuring that the most environmentally favorable reuse pathways appear at the top. Typically, we return the top-5 results by default, but the user can specify any subset size or additional filters as needed.

4.5.4 LLM Response Composition

Finally, a concise LLM generated response is displayed to the user. Because LLM output is tightly grounded in knowledge graph data (rather than raw text fragments), hallucination is greatly reduced. Each recommendation can be traced back to its matching entities, classification codes, and GWP100 values on the RDF graph, promoting transparency and reproducibility.

If the knowledge graph contains no relevant entries or mappings for the user's query, the system returns a brief message indicating that no matches were found in the database. However, the pipeline then relies on the broader knowledge of the LLM to provide any contextual information available, acknowledging that these details are not drawn from the verified graph. This fallback approach ensures that the user receives at least a partial or approximate answer, while clearly distinguishing graph-based facts from the LLM's more general training data. Users can refine the query or add

Algorithm 1 Knowledge Graph-based Retrieval and Reasoning Algorithm

Require: A natural language question Q
Ensure: A concise answer to Q
Step 1: Template Matching (LLM-Reasoning)
Use an LLM to parse Q and identify relevant entities (e.g., codes, resource or provider names), along with the intended search objective.
MatchedTemplates $\leftarrow \{\}$
for all `SPARQL template` T in **TemplateLibrary do**
  if `LLM semantic matching` confirms T fits Q's intent **then**
    MatchedTemplates $\leftarrow$ **MatchedTemplates** $\cup \{(\mathit{templateID} = T,\ \mathit{inputFields\ from}\ Q,\ \mathit{expectedOutput})\}$
  end if
end for

Step 2: Query Merging (LLM-Reasoning)
If multiple templates are selected, use an LLM to determine the optimal merging methods (e.g., intersection, chaining, union) based on Q's requirements.
Combine all templates in **MatchedTemplates** into a single SPARQL query **FinalQuery** (e.g., $A \cap B$, $A \to B$, or $A \cup B$).

Step 3: Execute and Retrieve Answer
Execute **FinalQuery** against the knowledge graph to obtain the result set $\mathbf{R}$.
Parse $\mathbf{R}$ to extract the entities or values requested by Q.
return a concise answer derived from $\mathbf{R}$.

additional details to improve the precision of future lookups and guide the continuous expansion of the knowledge graph coverage.

4.6 Experimental Setup

4.6.1 Hardware and Environment

All experiments are conducted on a Linux server equipped with an NVIDIA 4090 GPU, dual Intel Xeon processors, and 256 GB of RAM. We host the knowledge graph using GraphDB, and handle SPARQL queries through `SPARQLWrapper`. Vector embeddings are managed by FAISS, while LLM inference is provided via the Groq API. To enable flexible deployments across different environments, we use Python's `dotenv` to store endpoint URLs and credentials.

4.6.2 Models Evaluated

We benchmark the following models:

- **Llama3 (70B)** [36]: A 70B-parameter transformer model with a 4,096-token context window.
- **Qwen-qwq (32B)** [37]: A 32B-parameter variant tailored for domain-oriented tasks.

- **DeepSeek-R1-Distill-Llama (70B)** [38]: A 70B-parameter Llama derivative fine-tuned on specialized corpora.

All three models share identical inference hyperparameters (`temperature` = 0.7, `top-p` = 0.9, `max_tokens` = 4096) unless otherwise specified.

4.6.3 Baselines and Proposed Method

We compare our proposed CircuGraphRAG system to two baselines:

1. **Standalone LLM**: The user query is directly passed to the model without any external retrieval step.
2. **Naive RAG**: Unstructured text segments are retrieved, rather than relying on SPARQL queries and a structured knowledge graph.

In contrast, CircuGraphRAG combines structured SPARQL-based retrieval with a knowledge graph to improve both factual precision and sustainability-focused relevance.

4.7 Evaluation Metrics

4.7.1 Query Types

1. Single-Hop Queries

Examines whether the system can accurately retrieve a single fact from the graph.

2. Multi-Hop Queries

Determines the capacity of the system to chain multiple codes, constraints, and industrial relationships for valid reuse pathways.

4.7.2 Accuracy

We report two types of accuracy to evaluate the consistency of multi-round outputs and the correctness of responses under different template matching settings.

1. Round-level Precision

Measures the average accuracy of the model output over a sequence of evaluation rounds (Figure 3b). It is defined as

$$\text{Accuracy}_{\text{round}} = \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}[y_t^{\text{output}} = y_t^{\text{true}}], \tag{1}$$

where T is the total number of rounds.

2. Template Matching Accuracy

Assesses whether the system generates correct outputs under different template matching settings (Table 1).

4.7.3 ROUGE-L Precision, Recall, and F1

We use ROUGE-L, a recall-oriented metric based on the Longest Common Subsequence (LCS), to evaluate the lexical overlap between the generated and reference answers.

1. ROUGE-L Precision

Measures how much of the generated answer matches the reference, relative to the generated length:

$$\text{Precision}_{\text{ROUGE-L}} = \frac{LCS(C, R)}{|C|}. \tag{2}$$

2. ROUGE-L Recall

Measures how much of the reference answer is captured in the generated answer:

$$\text{Recall}_{\text{ROUGE-L}} = \frac{LCS(C, R)}{|R|}. \tag{3}$$

3. ROUGE-L F1 Score

Is the harmonic mean of ROUGE-L precision and recall:

$$\text{F1}_{\text{ROUGE-L}} = \frac{(1+\beta^2) \cdot \text{Precision}_{\text{ROUGE-L}} \cdot \text{Recall}_{\text{ROUGE-L}}}{\text{Precision}_{\text{ROUGE-L}} + \beta^2 \cdot \text{Recall}_{\text{ROUGE-L}}}. \tag{4}$$

where $\beta = 1$ by default. Here, $LCS(C, R)$ denotes the length of the longest common subsequence between the candidate answer C and the reference answer R.

4.7.4 BERTScore

BERTScore evaluates the semantic similarity between a generated answer $C = \{c_1, \ldots, c_m\}$ and a reference answer $R = \{r_1, \ldots, r_n\}$ by computing pairwise cosine similarities between their contextual embeddings. The final score is

$$\text{BERTScore}(C, R) = \frac{1}{2}\left(\frac{1}{|C|}\sum_{c_i \in C}\max_{r_j \in R}\cos(\vec{c}_i, \vec{r}_j) + \frac{1}{|R|}\sum_{r_j \in R}\max_{c_i \in C}\cos(\vec{r}_j, \vec{c}_i)\right), \tag{5}$$

where $\cos(\vec{c}_i, \vec{r}_j)$ denotes the cosine similarity between the contextual embeddings of the tokens c_i and r_j.

4.7.5 Exact Match Rate

EM measures the strictest form of matching, evaluating whether the predicted answer exactly matches the reference answer after normalization (lowercasing, removing articles and punctuation, etc.).

$$\text{EM} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\hat{y}_i = y_i], \tag{6}$$

where $\hat{y}_i$ is the predicted answer, y_i is the reference answer, $\mathbb{1}[\cdot]$ is the indicator function and N is the total number of evaluation instances.
By presenting the query process in a knowledge graph and ranking synergies by GWP100, CircuGraphRAG improves both the factual precision and environmental relevance of "waste-to-resource" recommendations, thus advancing data-driven sustainability in industrial ecosystems.

5 Conclusion

We introduced CircuGraphRAG, a knowledge graph based RAG framework that unites knowledge graph reasoning with LLMs to address waste-to-resource challenges in the circular economy. By mapping diverse industrial codes (EWC, NACE, ISIC, SSIC) and embedding regulatory metadata into a unified graph, CircuGraphRAG consistently surpassed baseline methods on single- and multi-hop queries, reducing hallucinations and boosting synergy discovery among potential waste producers and reusers.

Our experiments showed clear gains in factual accuracy, improved multi-hop reasoning, and lowered token usage when retrieving only relevant subgraphs. However, the effectiveness of CircuGraphRAG relies on continuously updated ontologies and access to real-time data, both of which remain open challenges. Future work will explore expanding domain coverage, incorporating dynamic sensor data, and collaborating with policymakers to encourage standardized data reporting. By systematically connecting industrial by-products to potential receivers, CircuGraphRAG marks a promising step toward more sustainable, circular production systems.

Declarations

- **Conflict of interest/Competing interests:** The authors declare that they have no conflict of interest.
- **Ethics approval and consent to participate:** Not applicable.
- **Consent for publication:** Not applicable.
- **Data availability:** Part of the data used in this study is obtained from the Ecoinvent dataset [35]. In compliance with data licensing terms, no actual numerical values from Ecoinvent are disclosed in this paper. The remaining datasets will be publicly released upon publication.
- **Materials availability:** Not applicable.
- **Code availability:** The relevant code for this study will be made publicly available after publication

References

[1] Chertow, M.R.: Industrial symbiosis: literature and taxonomy. Annual review of energy and the environment **25**(1), 313–337 (2000)

[2] Lombardi, D.R., Laybourn, P.: Redefining industrial symbiosis: Crossing academic–practitioner boundaries. Journal of industrial ecology **16**(1), 28–37 (2012)

[3] Püchel, L., Wang, C., Buhmann, K., Brandt, T., Schweinitz, F., Edinger-Schons, L.M., Brocke, J., Legner, C., Teracino, E., Mardahl, T.D.: On the pivotal role of data in sustainability transformations. Business & Information Systems Engineering **66**(6), 831–848 (2024) https://doi.org/10.1007/s12599-024-00904-4

[4] Solaimani, S.: From compliance to capability: On the role of data and technology in environment, social, and governance. Sustainability **16**(14) (2024) https://doi.org/10.3390/su16146061

[5] Nookala, G., Gade, K.R., Dulam, N., Thumburu, S.K.R., et al.: Governance for data ecosystems: Managing compliance, privacy, and interoperability (2024)

[6] Kullmann, F., Markewitz, P., Stolten, D., Robinius, M.: Combining the worlds of energy systems and material flow analysis: a review. Energy, Sustainability and Society **11**(1), 13 (2021) https://doi.org/10.1186/s13705-021-00289-2

[7] Moriguchi, Y., Hashimoto, S.: Material Flow Analysis and Waste Management, pp. 247–262 (2016). https://doi.org/10.1007/978-3-319-20571-7_12

[8] Moraga, G., Huysveld, S., Mathieux, F., Blengini, G.A., Alaerts, L., Van Acker, K., de Meester, S., Dewulf, J.: Circular economy indicators: What do they measure? Resources, Conservation and Recycling **146**, 452–461 (2019) https://doi.org/10.1016/j.resconrec.2019.03.045

[9] Roy, T., Garza-Reyes, J.A., Kumar, V., Kumar, A., Agrawal, R.: Redesigning traditional linear supply chains into circular supply chains–a study into its challenges. Sustainable Production and Consumption **31**, 113–126 (2022) https://doi.org/10.1016/j.spc.2022.02.004

[10] Tsang, A., Frost, T., Cao, H.: Environmental, social, and governance (esg) disclosure: A literature review. The British Accounting Review **55**(1), 101149 (2023)

[11] Hedberg, C.-J., Von Malmborg, F.: The global reporting initiative and corporate sustainability reporting in swedish companies. Corporate social responsibility and environmental management **10**(3), 153–164 (2003)

[12] Hales, J.: Sustainability accounting standards board (sasb). In: World Scientific Encyclopedia of Climate Change: Case Studies of Climate Risk, Action, and

Opportunity Volume 3, pp. 37–41. World Scientific, Singapore (2021)

[13] Board, F.S.: Task force on climate-related financial disclosures. Final Report: Recommendations of the Task Force on Climate-Related Financial Disclosures (2017)

[14] Fraccascia, L., Yazan, D.M.: The role of online information-sharing platforms on the performance of industrial symbiosis networks. Resources, conservation and recycling **136**, 473–485 (2018)

[15] Fraccascia, L., Giannoccaro, I.: What, where, and how measuring industrial symbiosis: A reasoned taxonomy of relevant indicators. Resources, conservation and recycling **157**, 104799 (2020)

[16] Capelleveen, G., Amrit, C., Zijm, H., Yazan, D.M., Abdi, A.: Toward building recommender systems for the circular economy: Exploring the perils of the european waste catalogue. Journal of environmental management **277**, 111430 (2021)

[17] Vidali, A., Jean, N., Pera, G.L.: Unlocking nace classification embeddings with openai for enhanced analysis and processing. arXiv preprint arXiv:2409.11524 (2024)

[18] Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., *et al.*: Knowledge graphs. ACM Computing Surveys (Csur) **54**(4), 1–37 (2021)

[19] Powers, S.: Practical RDF: Solving Problems with the Resource Description Framework. " O'Reilly Media, Inc.", Sebastopol, CA (2003)

[20] Lassila, O.: Resource description framework (rdf) model and syntax (1997)

[21] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)

[22] Rillig, M.C., Ågerstrand, M., Bi, M., Gould, K.A., Sauerland, U.: Risks and benefits of large language models for the environment. Environmental Science & Technology **57**(9), 3464–3466 (2023)

[23] Zhu, J.-J., Jiang, J., Yang, M., Ren, Z.J.: Chatgpt and environmental research. Environmental Science & Technology **57**(46), 17667–17670 (2023)

[24] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., Fung, P.: Survey of hallucination in natural language generation. ACM computing surveys **55**(12), 1–38 (2023)

[25] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., *et al.*: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing

Systems **33**, 9459–9474 (2020)

[26] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H.: Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997 (2023)

[27] Hu, Y., Lei, Z., Zhang, Z., Pan, B., Ling, C., Zhao, L.: Grag: Graph retrieval-augmented generation. arXiv preprint arXiv:2405.16506 (2024)

[28] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., Wang, H.: Retrieval-Augmented Generation for Large Language Models: A Survey (2024). https://arxiv.org/abs/2312.10997

[29] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (2021). https://arxiv.org/abs/2005.11401

[30] Kang, M., Kwak, J.M., Baek, J., Hwang, S.J.: Knowledge Graph-Augmented Language Models for Knowledge-Grounded Dialogue Generation (2023). https://arxiv.org/abs/2305.18846

[31] Godal, O., Fuglestvedt, J.: Testing 100-year global warming potentials: impacts on compliance costs and abatement profile. Climatic Change **52**(1), 93–127 (2002)

[32] United States Environmental Protection Agency: Understanding Global Warming Potentials | US EPA. Accessed: 2025-05-18 (2023). https://www.epa.gov/ghgemissions/understanding-global-warming-potentials

[33] Han, H., Wang, Y., Shomer, H., Guo, K., Ding, J., Lei, Y., Halappanavar, M., Rossi, R.A., Mukherjee, S., Tang, X., He, Q., Hua, Z., Long, B., Zhao, T., Shah, N., Javari, A., Xia, Y., Tang, J.: Retrieval-Augmented Generation with Graphs (GraphRAG) (2025). https://arxiv.org/abs/2501.00309

[34] Peng, B., Zhu, Y., Liu, Y., Bo, X., Shi, H., Hong, C., Zhang, Y., Tang, S.: Graph Retrieval-Augmented Generation: A Survey (2024). https://arxiv.org/abs/2408.08921

[35] Frischknecht, R., Rebitzer, G.: The ecoinvent database system: a comprehensive web-based lca database. Journal of Cleaner Production **13**(13), 1337–1343 (2005) https://doi.org/10.1016/j.jclepro.2005.05.002 . Life Cycle Assessment

[36] Meta AI: Introducing Meta Llama 3: The most capable openly available LLM. https://ai.meta.com/blog/meta-llama-3/. Accessed: 2025-05-18 (2024)

[37] Qwen Team: QwQ-32B: Embracing the Power of Reinforcement Learning. https:

//qwenlm.github.io/blog/qwq-32b/. Accessed: 2025-05-18 (2024)

[38] DeepSeek AI: DeepSeek-R1-Distill-Llama-70B. https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B. Accessed: 2025-05-18 (2025)

6 Appendix

6.1 Query–Answer Pairs from the Ground Truth.

Table 2: Query–Answer Pairs from the Ground Truth.

Case	Type	Query	Answer
1	Single-Hop	*Q: Find out the resource that is coded with EWC code 080121 and HS code 810330.*	A: Waste paint
2	Single-Hop	*Q: Find out the provider that is coded with NACE code 3821 and produces waste cement.*	A: Municipal incineration of waste polyethylene
3	Single-Hop	*Q: Find out the least value of GWP100 among the receivers coded with NACE code 3822.*	A: 0.008826959
4	Multi-Hop	*Q: The resources generated by the providers under the NACE code 3821 are all categorized under what CPA code and category?*	A: 382150; "Pellets of municipal waste"
5	Multi-Hop	*Q: Among the receivers of resources coded with CPA code 382150, which can be matched again with the receiver "market for waste polystyrene" as a provider?*	A: "treatment of waste polystyrene terephthalate, municipal incineration"
6	Multi-Hop	*Q: Among the providers of resources coded with EWC code 070213, which can also be a receiver of the "aluminium scrap"?*	A: "treatment of wastewater from PV cell production, wastewater treatment"

6.2 Single-hop QA performance for different LLMs and retrieval methods across 3 test cases.

Table 3: Single-hop QA performance for different LLMs and retrieval methods across 3 test cases.

Case	Method	LLM	ROUGE-L		
			Precision	Recall	F1
1	Standalone LLM	llama	0.0161	0.5	0.0313
		qwen	0.0045	0.5	0.0090
		deepseek	0.0086	0.5	0.0169
	Naive RAG	llama	0	0	0
		qwen	0	0	0
		deepseek	0	0	0
	CircuGraphRAG	llama	0	0	0
		qwen	1	1	1
		deepseek	1	1	1
2	Standalone LLM	llama	0.0165	0.4	0.0317
		qwen	0.0064	0.4	0.0127
		deepseek	0.0213	0.4	0.0404
	Naive RAG	llama	0.0099	0.4	0.0192
		qwen	0.0150	0.4	0.0290
		deepseek	0.0230	0.4	0.0435
	CircuGraphRAG	llama	0.4286	0.6	0.5
		qwen	0.4286	0.6	0.5
		deepseek	0.0909	0.2	0.125
3	Standalone LLM	llama	0	0	0
		qwen	0.0025	0.5	0.0050
		deepseek	0	0	0
	Naive RAG	llama	0	0	0
		qwen	0	0	0
		deepseek	0	0	0
	CircuGraphRAG	llama	1	1	1
		qwen	1	1	1
		deepseek	0	0	0

6.3 Multi-hop QA performance for different LLMs and retrieval methods across 3 test cases.

Table 4: Multi-hop QA performance for different LLMs and retrieval methods across 3 test cases.

Case	Method	LLM	ROUGE-L		
			Precision	Recall	F1
4	Standalone LLM	llama	0	0	0
		qwen	0	0	0
		deepseek	0	0	0
	Naive RAG	llama	0	0	0
		qwen	0	0	0
		deepseek	0	0	0
	CircuGraphRAG	llama	0	0	0
		qwen	1	1	1
		deepseek	0	0	0
5	Standalone LLM	llama	0.0273	0.4286	0.0513
		qwen	0.0385	0.2857	0.0678
		deepseek	0.0303	0.4286	0.0566
	Naive RAG	llama	0.0160	0.2857	0.0303
		qwen	0.0207	0.4286	0.0395
		deepseek	0.0417	0.4286	0.0759
	CircuGraphRAG	llama	1	1	1
		qwen	1	1	1
		deepseek	0.3158	0.8517	0.4615
6	Standalone LLM	llama	0.0165	0.2222	0.0308
		qwen	0.0069	0.3333	0.0135
		deepseek	0.0179	0.2222	0.0331
	Naive RAG	llama	0.0095	0.1111	0.0175
		qwen	0.0085	0.1111	0.0157
		deepseek	0.0357	0.2222	0.0615
	CircuGraphRAG	llama	0	0	0
		qwen	1	1	1
		deepseek	0	0	0

6.4 Single-Hop Timing and Token Usage.

Table 5: Single-Hop Timing and Token Usage.

Case	Method	LLM	Input Tokens	Output Tokens	Inference Time (s)
1	Standalone LLM	llama	30	106	0.39
	Standalone LLM	qwen	38	10572	25.82
	Standalone LLM	deepseek	23	707	3.37
	Naive RAG	llama	396	178	0.66
	Naive RAG	qwen	555	1826	4.23
	Naive RAG	deepseek	389	699	3.35
	CircuGraphRAG	llama	1140	66	0.27
	CircuGraphRAG	qwen	1220	789	1.97
	CircuGraphRAG	deepseek	1100	1570	5.80
2	Standalone LLM	llama	28	159	0.46
	Standalone LLM	qwen	30	3569	8.77
	Standalone LLM	deepseek	21	995	5.34
	Naive RAG	llama	314	258	0.89
	Naive RAG	qwen	420	646	1.48
	Naive RAG	deepseek	307	681	3.03
	CircuGraphRAG	llama	1508	71	0.30
	CircuGraphRAG	qwen	1206	1787	4.45
	CircuGraphRAG	deepseek	1192	2450	9.01
3	Standalone LLM	llama	30	304	0.98
	Standalone LLM	qwen	34	4075	10.01
	Standalone LLM	deepseek	23	1187	4.75
	Naive RAG	llama	319	94	0.31
	Naive RAG	qwen	426	1434	4.27
	Naive RAG	deepseek	312	1515	5.54
	CircuGraphRAG	llama	1297	82	0.36
	CircuGraphRAG	qwen	1408	1355	3.44
	CircuGraphRAG	deepseek	1377	1692	6.24

CircuGraphRAG input, output, and inference time are aggregated across three sequential stages: Template Matching (TM), Query Merging (QM), and Querying with Retrieved Context (QRC). For example, for Case 1–llama: 314 (TM) + 410 (QM) + 416 (QRC) = 1140 input tokens.

6.5 Multi-Hop Timing and Token Usage.

Table 6: Multi-Hop Timing and Token Usage.

Case	Method	LLM	Input Tokens	Output Tokens	Inference Time (s)
4	Standalone LLM	llama	32	135	0.45
	Standalone LLM	qwen	34	3764	8.58
	Standalone LLM	deepseek	25	968	4.35
	Naive RAG	llama	398	213	0.92
	Naive RAG	qwen	547	1149	2.89
	Naive RAG	deepseek	391	1251	5.19
	CircuGraphRAG	llama	1240	76	0.36
	CircuGraphRAG	qwen	1322	4719	11.50
	CircuGraphRAG	deepseek	1153	2434	8.92
5	Standalone LLM	llama	44	156	0.63
	Standalone LLM	qwen	48	8677	21.35
	Standalone LLM	deepseek	37	969	4.27
	Naive RAG	llama	344	160	0.63
	Naive RAG	qwen	449	1487	3.65
	Naive RAG	deepseek	337	1043	4.43
	CircuGraphRAG	llama	1356	79	0.29
	CircuGraphRAG	qwen	1506	1227	3.11
	CircuGraphRAG	deepseek	1409	1230	4.62
6	Standalone LLM	llama	37	168	0.55
	Standalone LLM	qwen	41	2172	5.30
	Standalone LLM	deepseek	30	944	4.20
	Naive RAG	llama	350	141	0.51
	Naive RAG	qwen	441	1000	2.30
	Naive RAG	deepseek	343	2230	8.13
	CircuGraphRAG	llama	1365	90	0.35
	CircuGraphRAG	qwen	1390	1373	3.47
	CircuGraphRAG	deepseek	1321	1646	6.34

CircuGraphRAG input, output, and inference time are aggregated across three sequential stages: Template Matching (TM), Query Merging (QM), and Querying with Retrieved Context (QRC). For example, for Case 4–llama: 374 (TM) + 468 (QM) + 398 (QRC) = 1240 input tokens.

6.6 Comparative Analysis of Token Usage and Inference Time Across the Three Stages of CircuGraphRAG (Cases 1–6)

Table 7: Comparative Analysis of Token Usage and Inference Time Across the Three Stages of CircuGraphRAG (Cases 1–6)

Case	LLM	Stage	Input Tokens	Output Tokens	Inference Time (s)
1	llama	Template Matching	314	56	0.17
		Query Merging	410	6	0.05
		Querying with Context	416	4	0.05
	qwen	Template Matching	334	481	1.13
		Query Merging	432	165	0.46
		Querying with Context	454	143	0.38
	deepseek	Template Matching	307	795	2.92
		Query Merging	384	475	1.75
		Querying with Context	409	300	1.13
2	llama	Template Matching	312	48	0.15
		Query Merging	755	6	0.08
		Querying with Context	441	17	0.07
	qwen	Template Matching	326	1120	2.77
		Query Merging	412	213	0.51
		Querying with Context	468	454	1.17
	deepseek	Template Matching	305	1635	5.98
		Query Merging	372	214	0.80
		Querying with Context	515	601	2.23
3	llama	Template Matching	338	69	0.22
		Query Merging	423	7	0.08
		Querying with Context	536	6	0.06
	qwen	Template Matching	360	597	1.51
		Query Merging	442	516	1.33
		Querying with Context	606	242	0.60
	deepseek	Template Matching	331	885	3.24
		Query Merging	392	479	1.76
		Querying with Context	654	328	1.24
4	llama	Template Matching	374	56	0.19
		Query Merging	468	5	0.06
		Querying with Context	398	15	0.11
	qwen	Template Matching	397	3691	9.05
		Query Merging	488	745	1.74
		Querying with Context	437	283	0.71
	deepseek	Template Matching	355	1275	4.66
		Query Merging	419	741	2.72

Continued on next page

Table 7 (continued)

Case	LLM	Stage	Input Tokens	Output Tokens	Inference Time (s)
		Querying with Context	379	418	1.54
5	llama	Template Matching	386	56	0.17
		Query Merging	480	6	0.05
		Querying with Context	490	17	0.07
	qwen	Template Matching	411	437	1.02
		Query Merging	500	360	0.97
		Querying with Context	595	430	1.12
	deepseek	Template Matching	379	483	1.79
		Query Merging	473	385	1.42
		Querying with Context	557	362	1.41
6	llama	Template Matching	380	75	0.23
		Query Merging	493	13	0.07
		Querying with Context	492	2	0.05
	qwen	Template Matching	405	764	1.92
		Query Merging	476	232	0.56
		Querying with Context	509	377	0.99
	deepseek	Template Matching	373	491	1.83
		Query Merging	489	561	2.10
		Querying with Context	459	594	2.41

6.7 Response Comparison for General Knowledge Question.

Table 8: Response Comparison for General Knowledge Question.

Question: What category of solid waste does waste polyurethane belong to?

Ground Truth: **Construction and demolition wastes**

Method	LLM	Response
Standalone LLM	**Llama**	Waste polyurethane (PU) is a type of synthetic polymer waste, and it belongs to the category of non-hazardous solid waste. Specifically, it falls under the category of **"Municipal Solid Waste" (MSW) or "Residual Waste."✗** In the European Union, waste polyurethane is classified under the European Waste Catalogue (EWC) code 17 04 07*, "Polyurethane waste." This code is part of the "Construction and Demolition Waste" (C&D waste) sector. In the United States, the Environmental Protection Agency (EPA) classifies waste polyurethane as a non-hazardous waste, and it is typically managed through municipal solid waste landfills or recycling facilities. It's worth noting that some polyurethane waste may be considered hazardous if it contains additives or contaminants—such as heavy metals, flame retardants, or volatile organic compounds (VOCs)—that render it hazardous. In such cases, the waste must be managed as hazardous waste under the applicable regulations and protocols.

(continued on next page)

Method	LLM	Response (continued)
	Qwen	<think> ... </think> Waste polyurethane is typically categorized within solid waste management systems based on its source and composition. Here's a structured breakdown: • **Municipal Solid Waste (MSW):** When derived from residential or commercial sources (e.g., discarded furniture, packaging, or consumer goods), polyurethane is generally classified as **municipal solid waste**. It falls under the "plastics" or "textiles/furniture" subcategory in MSW systems, depending on its form (e.g., foam, coatings, or rigid parts). • **Industrial Waste:** If generated during manufacturing processes (e.g., production scraps or offcuts), it may be categorized as **industrial waste**, subject to regulations governing industrial byproducts. • **Construction & Demolition (C&D) Debris:** When part of construction materials like insulation, roofing, or building components, it is classified as **C&D debris**, a distinct category from household waste. • **Hazardous Waste:** If the polyurethane contains hazardous additives (e.g., certain flame retardants, isocyanates, or other chemicals that make it toxic, reactive, or flammable), it may qualify as **hazardous waste** under regulations like the U.S. EPA's RCRA or the EU's Waste Framework Directive. This requires specialized handling. • **Recyclable/Recoverable Waste:** If being diverted for recycling (e.g., into new materials or energy recovery), it may be categorized separately as **recyclable or recoverable waste**, depending on local recycling infrastructure and policies. **Summary:** **Primary Category:** **Municipal solid waste (non-hazardous)** ✗ when from households or businesses. **Exceptions:** **Hazardous waste** ✗ if contaminated with toxic additives; industrial or C&D debris depending on origin. This classification ensures proper disposal, recycling, or regulation ....

(continued on next page)

Method	LLM	Response (continued)
	Deepseek	<think> . . . </think> Waste polyurethane is typically categorized as either **municipal solid waste** or **industrial solid waste**, depending on its source. • **Municipal Solid Waste**: This applies when polyurethane waste comes from consumer products, such as discarded furniture foam or household items. These items are often sent to landfills as part of general municipal waste. • **Industrial Solid Waste**: This applies when the waste is generated from manufacturing processes, such as production scraps or leftover materials from industrial applications. In cases where polyurethane is contaminated with hazardous substances, it may be classified as **hazardous waste ✗**. However, polyurethane itself is generally non-hazardous. Disposal methods often involve landfilling, though recycling is not always feasible due to the material's properties.
Naive RAG	**Llama**	**Retrieved Triplets:** Top results: 1: 3228870b-72a1. . . -Resource, hasInformation, 'waste polyurethane' is a solid waste under the category of construction and demolition wastes.. . . . (Distance: 0.4850251078605652) *(Note: All LLMs use the same retrieved context information.)* Based on the retrieved contexts, the answer to the question is that **waste polyurethane belongs to the category of construction and demolition wastes ✓**. It can also belong to the following waste categories: **industrial**.
	Qwen	<think> Okay, let's see. The user is asking which category of solid waste waste polyurethane belongs to. . . . The answer is that it's under construction and demolition wastes, and can also be industrial. </think> The waste polyurethane is categorized as a **solid waste under the category of construction and demolition wastes ✓**. Additionally, depending on its source, it may also fall under the **industrial waste category**. This classification is based on its generation during production and construction activities across various sectors, as stated consistently across all retrieved contexts.

(continued on next page)

Method	LLM	Response (continued)
	Deepseek	<think> Okay, I need to figure out what category of solid waste waste polyurethane belongs to. . . . It's non-hazardous and has specific disposal methods like landfills and incineration. The contexts are consistent across all five, so I can confidently state the category based on this information. </think> Waste polyurethane belongs to the category of **construction and demolition wastes** ✓ and can also be classified under industrial waste categories. It is considered non-hazardous and is typically disposed of through methods such as landfill deposition or municipal incineration.
CircuGraphRAG	**Llama**	**1. Template Matching:** [{"query": 17, "input": "waste polyurethane", "output": "category of solid waste"}, {"query": 6, "input": "waste polyurethane", "output": "resource code"}] **2. Query Merging:** 17 **3. Querying with Retrieved Context:** **construction and demolition wastes, industrial** ✓
	Qwen	**1. Template Matching:** <think> . . . </think> - Possible Solutions: [{"query": 17, "input": "waste polyurethane", "output": "category information"}] **2. Query Merging:** <think> . . . </think> 17 **3. Querying with Retrieved Context:** <think> . . . </think> **construction and demolition wastes** ✓

(continued on next page)

Method	LLM	Response (continued)
	Deepseek	**1. Template Matching:** **<think>** ... **</think>** [{"query": 17, "input": "waste polyurethane", "output": "category of solid waste"}] **2. Query Merging:** **<think>** ... **</think>** Since the question directly asks for the category of solid waste for "waste polyurethane," and query 17 is specifically designed to retrieve information based on the entity's name, the solution is to execute query 17 with the input "waste polyurethane." **Merging method**: 17("waste polyurethane") **3. Querying with Retrieved Context:** **<think>** ... **</think>** The category of solid waste that waste polyurethane belongs to is **construction and demolition wastes ✓**, and it can also be classified as industrial waste.

Note: For brevity, the `<think>` sections are partially omitted in this table.

6.8 Response Comparison for Multi-hop Question.

Table 9: Response Comparison for Multi-hop Question.

Question: The resources generated by the providers under the NACE code 3821 are all categorized under what CPA code?
Ground Truth: **382150**

Method	LLM	Response
Standalone LLM	**Llama**	According to the European Commission's website, the NACE code 3821 corresponds to **"Renting of air transport equipment"** and is categorized under CPA (Common Procurement Vocabulary) code **63500000 ✗**, which covers **"Air transport services"**. **Breakdown:** • **NACE code 3821:** Renting of air transport equipment • **CPA code 63500000:** Air transport services
	Qwen	**<think>** Okay, let me try to figure out the answer to this question. ... Therefore, the resources (products) from NACE 3821 would be under CPA code 3111. **</think>** The resources generated by providers under the NACE code **38.21** ("Manufacture of semiconductors and discrete components") are categorized under **CPA code 3111** in the Classification of Products by Activity (CPA). **Breakdown:** - **NACE 38.21** → Production of semiconductors and discrete electronic components. - **CPA 31** → "Manufacture of computer, electronic and optical products". - **CPA 3111** → "Semiconductors and discrete components". **Thus, the CPA code for these resources is 3111 ✗**. This ensures consistency between the economic activity and its product classification. **Note:** Always verify with the latest CPA version, as codes may be updated.

(continued on next page)

Method	LLM	Response (continued)
	Deepseek	<think>Okay, so I need to figure out the CPA code that categorizes the resources generated by providers under the NACE code 3821. I'm not entirely sure what NACE and CPA codes are, but I think they're some kind of classification systems used in the EU. . . . Yes, I think that's correct. So, the resources (services) from NACE 3821 are categorized under CPA code 37.10.00. </think> The resources generated by providers under the NACE code 3821, which pertains to waste collection (including both hazardous and non-hazardous waste), are categorized under the CPA code 37.10.00. This CPA code specifically covers waste collection services. **Answer: CPA code 37.10.00 ✗**
Naive RAG	**Llama**	**Retrieved Triplets:** `1: 37f20209-b2a3-4950-ac0d-88c830c73c2c, hasResource, 37f20209-b2a3-4950-ac0d-88c830c73c2c_Resource (Distance: 1.03)` . . . *(Note: All LLMs use the same retrieved context information.)* **Response:** Based on the provided context, there is no direct information about NACE codes or CPA codes. The context only mentions NLP derived EWC, Translated CPA, and NLP derived HScode, but it does not provide any specific values or mappings between them. Therefore, I'll have to rely on my own knowledge. NACE (Nomenclature of Economic Activities) is a European classification system for industries and economic activities. CPA (Classification of Products by Activity) is a classification system for products. **Unfortunately, without more specific information, it's not possible to determine the exact CPA code corresponding to NACE code 3821 ✗**. NACE code 3821 corresponds to "Manufacture of bicycles and invalid carriages", but the CPA codes for products related to this industry can vary depending on the specific product. If you have more information or context about the resources generated by the providers, I might be able to help you with the CPA code categorization.

(continued on next page)

Method	LLM	Response (continued)
	Qwen	**<think>** Okay, let's see. The user is asking about the CPA code that categorizes resources from providers with the NACE code 3821. . . . Since the user is expecting an answer, maybe the intended answer is CPA 24. I'll go with that, but note that the given contexts don't explicitly state it. **</think>** The resources retrieved do not explicitly mention NACE code 3821 or any CPA code in their triplet structures. However, based on standard classification mappings between NACE and CPA codes, **NACE code 38.21 ("Manufacture of office, accounting, and computing machinery")** typically corresponds to CPA code 24.11 ("Computing machinery and software"). If forced to infer from the closest context (the first triplet, with the shortest distance of 1.0286), the answer might align with common categorizations, but this is speculative. The provided contexts lack explicit links between NACE 3821 and CPA codes, so the answer is best approximated using external knowledge of standard classifications. **Answer:** The resources under NACE code 3821 are generally categorized under **CPA code 24.11** ✗. However, this requires confirmation from the actual resource data linked in the graph database.
	Deepseek	**<think>** Okay, I'm trying to figure out the answer to the question: "The resources generated by the providers under the NACE code 3821 are all categorized under what CPA code?". Let me break this down step by step. First, I know that NACE codes are used for classifying businesses in the European Union. Specifically, NACE code 3821 refers to "Waste treatment and disposal". So, the question is asking about resources generated by providers in this waste treatment sector. . . . However, without specific CPA code numbers from the context, I can't provide an exact code. The context only mentions that CPA codes are translated from NLP outputs, but doesn't specify which ones. Therefore, based on my own knowledge, I can infer that the CPA codes for resources under NACE 3821 would be within the waste treatment and disposal category, possibly starting with 38. But I can't provide the exact code without more specific information. **</think>** **Response:** The resources generated by providers under the NACE code 3821, which pertains to waste treatment and disposal, are categorized under CPA codes related to waste management. Specifically, these resources fall under **CPA code 38.00.00** ✗, which covers "Waste treatment and disposal."

(continued on next page)

Method	LLM	Response (continued)
CircuGraphRAG	**Llama**	**1. Template Matching:** [{"query": 10, "input": "NACE code 3821", "output": "resource name"}, {"query": 5, "input": "resource name", "output": "CPA code"}] **2. Query Merging:** **Merging method:** Query 10 ⇒ Query 5 **3. Querying with Retrieved Context:** The SPARQL query returned no results. The providers under NACE code 3821 are likely related to waste treatment, so the resources generated might be categorized under **CPA code 38 (Waste collection, treatment and disposal activities; materials recovery) ✗**.
	Qwen	**1. Template Matching:** **<think>** . . . **</think>** • Query 1: Input = NACE 3821 → Output = Providers under that code • Query 7: Input = Provider names → Output = Resources generated by providers • Query 6: Input = Resource names → Output = CPA codes (resource codes) **2. Query Merging:** **<think>** . . . **</think>** **Merging Strategy:** Chaining **Execution Flow:** Query 1 ⇒ Query 7 ⇒ Query 6 **3. Querying with Retrieved Context:** **<think>** . . . **</think>** The SPARQL query returned the code cpa: 382150, hence the CPA code that categorizes the resources generated by providers under NACE 3821 is **382150 ✓**.

(continued on next page)

Method	LLM	Response (continued)
	Deepseek	**1. Template Matching:** **<think>** ... **</think>** - Possible Solutions: [{"query": 18, "input": "NACE code 3821", "output": "provider information (including resources)"}, {"query": 5, "input": "resource code(s) from provider", "output": "CPA code"}] **2. Query Merging:** **<think>** ... **</think>** The merging method is chaining because the output of query 18 (resource codes) is used as the input for query 5 to get the CPA codes. **Merging method:** Query 18 ⇒ Query 5 **3. Querying with Retrieved Context:** **<think>** ... **</think>** The SPARQL query returned no results. Based on general knowledge, resources under NACE code 3821 (waste collection, treatment, and disposal activities) are typically categorized under CPA code 37.20 (Waste collection, treatment, and disposal activities). Answer: **CPA code 37.20 ✗**.

Note: For brevity, the `<think>` sections are partially omitted in this table.

6.9 List of Abbreviations.

Table 10: List of Abbreviations.

Abbreviation	Full Name	Explanation
LLM	Large Language Model	AI model trained on large corpora for text generation.
LCS	Longest Common Subsequence	The longest ordered subsequence shared by two sequences, allowing gaps but preserving order.
RAG	Retrieval-Augmented Generation	Combines an LLM with external data or knowledge bases to reduce hallucinations.
CircuGraphRAG	Circular Graph-based RAG	Proposed system anchoring the LLM in a waste-to-resource knowledge graph.
GraphRAG	Graph-based RAG	RAG that uses knowledge graphs for multi-hop queries.
ESG	Environmental, Social, and Governance	Criteria for evaluating organizations' sustainability and societal impact.
EWC	European Waste Catalogue	EU classification system for listing and categorizing waste.
NACE	Nomenclature of Economic Activities	EU classification of economic activities.
CPA	Classification of Products by Activity	Connects products to the economic activity that produces them.
HS	Harmonized System	Globally standardized codes for classifying traded goods.
ISIC	International Standard Industrial Classification	UN classification of global economic activities.
WZ	Wirtschaftszweige	German classification of economic activities.
SSIC	Singapore Standard Industrial Classification	Singapore's national standard for economic activities.
TCFD	Task Force on Climate-related Financial Disclosures	Framework for climate-related financial transparency.
GRI	Global Reporting Initiative	Guidelines for sustainability and social responsibility reporting.
SASB	Sustainability Accounting Standards Board	Industry-specific sustainability disclosure standards.
RDF	Resource Description Framework	W3C standard for implementing knowledge graphs.

(continued on next page)

Abbreviation	**Full Name**	**Explanation**
GWP100	100-year Global Warming Potential	A gas's heat-trapping impact over 100 years vs. CO_2.
PU	Polyurethane	Synthetic polymer used in foams and coatings.
C&D	Construction and Demolition	Waste generated by construction/demolition activities.
EPA	Environmental Protection Agency	U.S. environmental regulatory authority.
VOC	Volatile Organic Compound	Quickly evaporating organic pollutants.
RCRA	Resource Conservation and Recovery Act	U.S. law on solid and hazardous waste disposal.
SPARQL	SPARQL Protocol and RDF Query Language	Query language for RDF-based knowledge graphs.
NLP	Natural Language Processing	AI for analyzing and generating human language.
Ecoinvent	*Ecoinvent Database*	Industrial life cycle database with resource and emissions data.
FAISS	Facebook AI Similarity Search	Library for vector similarity search and clustering.
RDFlib	RDF Library	Python library for RDF graph handling.
SPARQLWrapper	SPARQLWrapper Library	Python tool for sending SPARQL queries to endpoints.
Groq (API)	Groq API	High-performance runtime for LLM inference.
MPNet	Masked and Permuted Pre-training	Transformer model for sentence embeddings.