SoftwareX 34 (2026) 102602

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Do you speak LCA? FAULDIER: A framework for large language model assisted Life Cycle Inventories in Life Cycle Assessment

Lukas Lazar

Institute for Technology Assessment and System Analysis (ITAS), Karlsruhe Institute of Technology (KIT), Karlstrasse 11, 76133 Karlsruhe, Germany

A R T I C L E I N F O

Keywords:
Life Cycle Assessment (LCA)
Large language model (LLM)
Environmental sustainability assessment

A B S T R A C T

Advances in Life Cycle Assessment (LCA) toward greater automation and methodological integration have intensified challenges in standardizing heterogeneous raw Life Cycle Inventory (LCI) data, which rarely aligns with LCI database nomenclature. Rule-based mapping approaches struggle with linguistic variations, typographical errors, unit inconsistencies, and location granularity mismatches. Furthermore, they fail to adapt automatically when data or terminology change. FAULDIER (Framework for lArge langUage modeL assisteD lIfe cyclE inventoRy) is proposed as a framework to bridge heterogeneities between raw LCI data and LCI database requirements. It aims to automate data transformation by resolving naming inconsistencies, classifying flow types, and harmonizing locations and units. By using LLMs, FAULDIER supports handling multilingual inputs, correcting typographical errors, resolving location granularity mismatches, and choosing proxies for missing processes. In a test scenario using the open LCI database FORWAST and a use case characterized by non-standardized multilingual entries, unit inconsistencies, and typographical errors, FAULDIER achieved approximately 57% process and elementary flow mapping accuracy (single-expert validated), with unit conversion error rates below 1%. Current limitations include LCI database constraints, LLM token limitations, performance variability of open-weight LLMs, mapping ability, and reproducibility across runs. Within these limitations, FAULDIER indicates the feasibility of LLM-assisted LCI construction for LCA modeling, particularly for non-standardized raw LCI data. Future work could focus on developing confidence metrics for mapped LCI data, optimizing LLM query efficiency, and expanding testing across additional LCI databases, use cases, and LLMs.

Metadata

This ancillary data table is required for the sub-version of the codebase. Please replace the italicized text in the right column with the correct information about your current code and leave the left column untouched.

Nr	Code metadata description	*Metadata*
C1	Current code version	*0.1.0*
C2	Permanent link to code/ repository used for this code version	*https://github.com/ljlazar/fauldier*
C3	Permanent link to reproducible capsule	https://mybinder.org/v2/gh/ljlazar/fauldier/HEAD?urlpath=%2Fdoc%2Ftree%2F%2Fexample%2FLCA_LLM.ipynb
C4	Legal code license	*BSD-3-Clause*
C5	Code versioning system used	*Git*
C6	Software code languages, tools and services used	*Python, SAIA GWDG*
C7	Compilation requirements, operating environments and dependencies	*jupyterlab, brightway2, IPython, pandas, numpy, thermo, openai, tiktoken,*
C8	If available, link to developer documentation/manual	
C9	Support email for questions	lukas.lazar@kit.edu

1. Motivation and significance

Life Cycle Assessment (LCA) has evolved through successive development stages, driven both by internal methodological progress and the incorporation of innovations from other disciplines. These developments have expanded its capabilities and enabled the resolution of emerging challenges. Today, LCA may be at the threshold of another major transformation, driven by the growing need for broader dissemination and the opportunities offered by Large Language Models (LLMs).

E-mail address: lukas.lazar@kit.edu.

https://doi.org/10.1016/j.softx.2026.102602
Received 24 September 2025; Received in revised form 17 February 2026; Accepted 8 March 2026
Available online 13 March 2026

The conceptual origins of LCA trace back to predecessor methods from the 1960s, including systems analysis, the life-cycle concept as a budgeting process for military weapon systems, and resource and environmental profile analysis by corporations [1–5]. Due to limited computing resources in this era [6], these early assessments were typically performed manually. Specialized LCA software (e.g. GaBi, SimaPro) emerged around 1990 [4], enabling more efficient handling of Life Cycle Inventory (LCI) data and facilitating the calculation of Life Cycle Impact Assessment (LCIA) results. However, early software capabilities were limited to only a few impact categories [7], thus restricting advanced functionalities such as scenario modeling, uncertainty analysis, and other automation-intensive tasks which require numerous calculation runs.

Throughout the 2010s and 2020s, Python-based frameworks such as Brightway [8], complemented by tools such as lca_algebraic [9] and premise [10], enabled ready-to-use, parametrized LCA that facilitated and accelerated prospective assessments with a multitude of scenarios. Prospective LCA relies, even more than standard LCA, on LCI data collection from non-specialists (e.g. scientific project partners, companies), who typically provide generic material and energy flow descriptors that rarely align with LCI database naming. Furthermore, coupling LCA with different complementary methods and tools (e.g. energy system modeling [11], CAD [12]) complicates data standardization, as heterogeneous outputs from these tools require manual formatting to align with LCI databases.

Rule-based data mapping approaches using logic are limited by their reliance on predefined rules, which are difficult to scale to large real-world domains [13,14] and to keep them up to date. In the context of LCA applications, this typically results in fixed input-output schemas, linking a single software tool to one LCI database version. Moreover, because these mapping approaches rely on classical logic (true/false statements), they cannot effectively resolve differences across languages, typographical errors, unit inconsistencies, or differences in location granularity, unless a rule is explicitly implemented for each case. While fuzzy logic [15–17] can partially address these issues, the acquired knowledge remains static [18], making it difficult to adapt to changes such as updates within the LCI database or switching to a different LCI database. Consequently, recent developments in other fields combine these methods with LLMs to overcome these limitations [19,20].

Since 2018, advances in LLMs have demonstrated significant capabilities in resolving such challenges, particularly through reasoning models [21]. Recent studies have identified promising applications of LLMs in LCA. These include supporting LCI data collection and interpreting LCIA results [22], enabling direct extraction of impact data from scientific literature [23], and, in the context of Social LCA, retrieving information from documents, articles, webpages, and similar sources. Furthermore, LLMs have also been applied to generate process flow graphs [24] and to estimate greenhouse gas emissions (e.g. Scope 3 emissions of a company) with a reported precision of around 87% [25]. Using an LLM-based agent, carbon footprint reports were automatically generated [26]. However, an expert-grounded benchmark revealed that 37% of LLM-generated responses in the LCA domain were inaccurate, highlighting significant risks [27]. Despite these risks, LLMs could serve as an effective bridge to reconcile heterogeneities between raw data and the requirements of LCA. Recent review studies on LCA automation using artificial intelligence conclude that, while such approaches can enable scalable and rapid decision-making in environmental sustainability assessment [28,29], only very few tools are currently publicly available [28]. To evaluate LLM capabilities in the context of LCA, within an open-source and LCA-software-integrated tool, FAULDIER (Framework for lArge langUage modeL assisteD lIfe cyclE inventoRy) was developed, a framework designed to address the following objectives:

1. Spreadsheet-to-LCA: Transform raw LCI data from a simplified spreadsheet into an LCA-compatible format, requiring exact nomenclature input.
2. LLM-assisted mapping: Map processes and elementary flows by resolving heterogeneities in process names, multilingual entries, unit variations, location specifications, and missing input data.
3. Integration into LCA software: Enable full automation from raw LCI data to LCIA results.

Compared to existing LLM-assisted LCA tools in research, FAULDIER focuses on raw LCI data typically encountered in research projects that require a prospective LCA approach. Consequently, its development emphasizes integration with flexible LCA software such as Brightway, which supports connections to additional tools for prospective LCA, LCI parametrization, and sensitivity and uncertainty analysis.

2. Software description

2.1. Software architecture and functionalities

The core function x2bw_transformation in FAULDIER automates the conversion of raw LCI data from the LCI spreadsheet into a Brightway-compatible format by:

– Default location assignment if none is provided
– Categorization of processes (e.g. electricity) into technosphere and elementary flows (e.g. carbon dioxide) to biosphere
– Optional avoided-burden allocation for multi-output processes

Configurable via basic_mapping.py, these transformations reduce the need for manual data curation. For non-standardized raw LCI data, llm_mapping.py can use LLMs to resolve:

– Naming inconsistencies: Mapping generic descriptors to database-specific process or elementary flow names
– Linguistic variations: Handling inputs in multiple languages
– Typographical errors: Correcting misspellings in process or elementary flow names
– Location granularity mismatches: Mapping locations to database-specific regions
– Proxy processes: Selecting substitutes for processes unavailable in the LCI database

The complete workflow (Fig. 1) is demonstrated in the Jupyter notebook LCA_LLM.ipynb, which:

1. Extracts processes and elementary flows from the LCI database
2. Transforms the LCI spreadsheet file (LCA_AP_LLM.xlsx) into a Brightway database
3. Maps process names from the LCI spreadsheet; if LLM_mapping=True, starts mapping via the LLM
4. Constructs the product system for the subsequent LCIA calculation

2.2. Software requirements

To use FAULDIER, Brightway must be installed [30] and initialized, including the installation of an LCI database (e.g. FORWAST). API access to an LLM is required and can be configured in the /input/llm_config. txt file, or will be requested upon first execution. Because results may differ across various LLMs, the prompt (/input/llm_prompt.txt) and LLM parameters may require adjustment when employing different models or data inputs.

2.3. Software design decisions

Although databases such as ecoinvent [31] are widely used in LCA

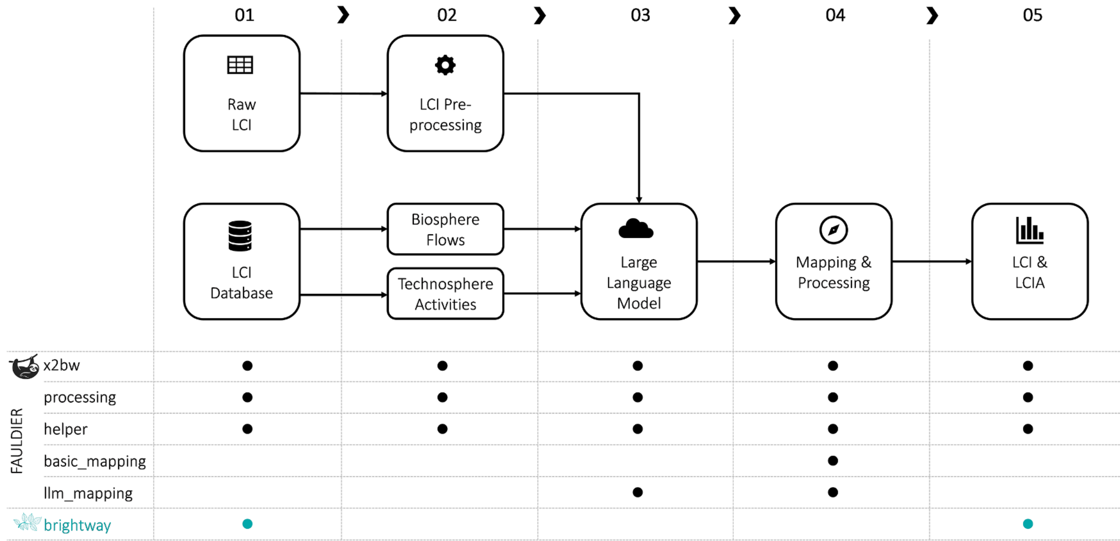

Fig. 1. Illustration of FAULDIER's main components, architecture, and information flow.

practice, we implemented the FORWAST [32] database. The ecoinvent license agreement [33] restricts the export of data, which likely includes the transmission to external LLMs. Furthermore, the ecoinvent database, including its elementary flows, would generate approximately 900,000 tokens, exceeding the maximum context window of all LLMs available to us for this study [34]. For example, using Alibaba Cloud Qwen 3 235B A22B Thinking 2507 LLM, which supports up to 222,000 tokens, we could transmit approximately one-quarter of the entire ecoinvent database in a single request. Thus, while FAULDIER was specifically designed with database flexibility in mind, the current implementation and testing were exclusively performed with the FORWAST database.

Since FAULDIER is designed to directly handle non-standardized raw LCI data, simplified heuristic decision rules were derived for the mapping. For example, generic inputs such as "Nitrogen" could represent an input elementary flow (natural resource), an output emission, or a process. To resolve this ambiguity, we require the environmental compartment for elementary flows in the spreadsheet and instruct the LLM to map to elementary flows whenever a compartment is provided. If neither a location nor a compartment is given, the LLM shall default to mapping the input to a process. After the LLM step, FAULDIER assigns all entries with environmental compartments to elementary flows. For by-products, a simplified avoided-burden approach is applied, wherein the by-product is included as a negative input. This can be overridden by entering "no avoided burden" in the "Description" field, which results in a cut-off. Thus, this approach does not cover the full range of LCA allocation methods.

Regarding LLM selection criteria, we used open-weight models that ensured data protection (compliant with GDPR) and with context windows exceeding 128,000 tokens, hosted by the Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG) [35,36]. Language- or application-specific models were excluded to ensure robust multilingual capability. Meeting these requirements, the following open-weight models were available to us: Meta Llama 3.1 8B Instruct, OpenAI GPT-OSS 120B, Google Gemma 3 27B Instruct, Alibaba Cloud Qwen 3 235B A22B Thinking 2507, Alibaba Cloud Qwen QwQ 32B, Meta Llama 3.3 70B Instruct, and Mistral Large Instruct.

3. Illustrative example

To demonstrate the functionality of FAULDIER, we implemented a test case using a simplified LCI for synthesis gas production via Artificial Photosynthesis. The LCI data were adapted from prior research [37] and modified to enable the use of the FORWAST database and to simulate real-world data challenges, including multilingual process names, non-standardized geographical locations, inconsistent units requiring conversion, and typographical errors. Consequently, the resulting LCI does not constitute a comprehensive LCA but serves exclusively as a demonstration of FAULDIER's mapping capabilities.

Using the LCA_LCI_LLM.xlsx raw LCI data input spreadsheet, we generated an LCA model by processing the test LCI data through multiple LLMs. Among the seven evaluated open-weight LLMs, Alibaba Cloud Qwen 3 235B A22B Thinking 2507 was the only model that successfully mapped all requested processes and elementary flows. The remaining models either failed to produce the required output format or left most inputs unmapped. We iteratively refined the prompt and experimented with different temperature and top_p settings, but full mapping could not be achieved with any other models.

Consequently, we analyzed three runs using Alibaba Cloud Qwen 3 235B A22B Thinking 2507 with temperature = 0.5 and top_p = 1. The mapping results for each input are provided in the Supplementary Information as mapping_results.xls. For some inputs, the mapped process or elementary flows varied between runs. This also occurred in the unit conversion for biogas, where the converted value differed across runs. When "Gas" was assumed as a proxy for biogas, the LLM did not account for differences in heating values between natural gas and biogas. Several elementary flows were misclassified, for example, carbon dioxide was mapped to the resource category (e.g. "carbon dioxide, from soil or biomass stock") rather than the correct emission category ("carbon dioxide, non-fossil"). Specific chemicals not available in the LCI database, including didodecyldimethylammonium bromide, toluene, acetone, hexane, and isopropanol, were mapped to the generic chemical production process. Ammonia and nitrogen were mapped to fertilizer. However, in some runs, nitrogen as well as gold, acetone, and isopropanol were mapped to elementary flows. Photocatalyst production facilities were mapped to "Machinery and equipment", "Buildings, non-residential", or "total production and consumption 2003", while the photocatalyst deposition facility was mapped to "Buildings, non-residential" or "Infrastructure, excluding buildings". All of these LCI inputs were labeled as "factory" in the source data. All electricity LCI inputs were successfully mapped despite containing special characters, multilingual entries (tested in German, Spanish, and Japanese), and spelling errors. However, similar to the factory mappings, the geographical location assigned to the electricity processes (e.g. DK/EU27) varied between runs using the same LLM.

From an LCA practitioner's perspective, acknowledging that interpretations may vary between practitioners, approximately 57% of processes were correctly mapped, 20% were incorrectly mapped, and the remainder were classified as inconclusive. For material flows provided in volumetric units, the LLM applied density-based conversions to express values in mass units. The resulting error remained below 1% when compared to standard reference values.

Running times varied significantly across LLMs. The Alibaba Cloud Qwen 3 235B A22B Thinking 2507 model required approximately 1 hour, whereas Meta Llama 3.1 8B Instruct, OpenAI GPT-OSS 120B, Google Gemma 3 27B Instruct, Alibaba Cloud Qwen QwQ 32B, Meta Llama 3.3 70B Instruct, and Mistral Large Instruct finished in under 10 minutes. This corresponds to approximately 17 s to 2 min per mapping of one process or elementary flow (35 in total). The complete mapping outputs with rationales for each decision of the LLM are documented for one run in the Supplementary Information (LLM_output.txt).

4. Discussion and limitations

4.1. Discussion of results

The incorrect mapping of elementary flows, such as assigning emissions to resources (or vice versa) or processes to elementary flows, represents a major issue. This occurs because the LLM does not consistently follow the predefined rules outlined in the design decisions, a phenomenon described as instructional distraction in LLM research [38]. This may stem from similar causes as hallucinations, occurring because acknowledging uncertainty is less rewarding for the model [39]. When configuration parameters (temperature and top_p) were adjusted, several processes remained unmapped. Multiple attempts to refine the prompt also failed to fully resolve this problem.

More complex and partly subjective tasks, such as converting biogas to natural gas based on heating value, were not performed. This limitation may be mitigated with greater data availability in the LCI database. Limited data availability likely also contributed to mappings such as biogas being mapped to “Gas” and specific chemicals being mapped to “Chemicals”. While this is not ideal, it aligns to some extent with common LCA practice, especially when there are limitations on modeling the missing process.

Inconsistent mappings such as assigning the parabolic trough collector to “Machinery and Equipment” in one run and to “Metals basic” in another indicate fundamental ambiguity in the LLM’s results. Similar inconsistencies were observed for different facility types (both labeled as “factory” in the input) and different process locations across multiple runs.

Unit mismatches introduce additional ambiguity. For instance, mapping the photocatalyst production facility to “Buildings, non-residential” introduces complications because the FORWAST database expresses this process in monetary units (EUR 2003), whereas other databases represent such facilities in physical units. This discrepancy illustrates the risk of ignoring unit inconsistencies during LLM-based mapping. Despite these limitations, the successful application of density-based conversions for volume-to-mass transformations shows the framework’s capability to convert units.

The mapping of CO_2 biogenic to “Carbon dioxide, non-fossil” reflects an appropriate mapping of biogenic carbon streams. The mapping of ammonia to “Fertilizer, N” demonstrates contextual reasoning but also introduces new challenges, such as determining the appropriate equivalent quantity, which was not solved by the LLM.

The LLM’s performance with electricity inputs containing special characters, multilingual entries, and intentional spelling errors demonstrates linguistic flexibility. Moreover, despite the previously noted misalignments, elementary flow mappings consistently achieved high accuracy across all test cases.

Currently, to the best of our knowledge, there are no studies that use LLMs on heterogeneous LCI data to automate LCA in a way that would allow for direct comparison to our results. However, there are similar tools and approaches: For example, the Flamingo tool applies natural language machine learning to automatically identify environmental impact factors for products [40]. On a dataset of 664 products, it achieved a matching accuracy of 75%. Furthermore, the Parakeet tool estimates greenhouse gas emissions (e.g. corporate scope) with an average precision of approximately 87% [25]. A whitepaper by Makersite evaluated LLM-based estimates for global warming potential and found that, for some products, the deviation from modeled impact assessment results was extremely high, up to 508,903% [41]. In contrast, a study in which an AI-powered agent calculated environmental impacts based on information extracted from webpages and text documents, reported a maximum deviation of approximately 12% compared to a full LCA [42].

Our evaluation was based on a single raw LCI dataset, which limits the validity of the results compared to testing across multiple datasets from diverse sources. However, by publishing the tool as open source, we aim to enable community-driven testing on a broader range of data, which could help strengthen and validate our findings.

4.2. Database limitations

A major limitation of this study is the inability to use the larger-scale ecoinvent LCI database, which is widely adopted by LCA practitioners. Consequently, we were unable to benchmark LLM-generated mappings against practitioner-derived mappings based on ecoinvent, substantially reducing the significance and comparability of our results. Due to FORWAST’s limited process availability and lower specificity, identifying suitable processes was inherently challenging, which may have negatively affected performance. However, FAULDIER is structurally designed to allow for database interchangeability, and the prompt can be adjusted to include database-specific requirements. Nevertheless, in accordance with good LCA practice, this functionality should be used only to transfer the entire product system to another database, because assumptions and methodological approaches differ across databases. Mixing databases may therefore lead to inconsistent results [43].

4.3. LLM limitations

Token consumption limitations present a significant technical barrier to full database integration. Our testing revealed that, even with the relatively limited FORWAST database, we approached a token capacity of approximately 98,000 tokens when transmitting name, reference product, location or environmental compartment, and unit of each process and elementary flow. Full integration of comprehensive databases like ecoinvent would only be feasible with commercial models allowing up to 1 million tokens, unless substantial filtering or segmentation of the database is applied. To reduce token size, compression tools such as LLMLingua [44] and pre-filtering methods (e.g. semantic-similarity pre-filtering [45]) should be evaluated, as LCI databases will likely continue to grow in size [46]. However, it is also likely that future LLM versions will support even higher token capacities, potentially mitigating this problem.

A significant risk of using LLMs in this context is incorrect mapping, i.e. assigning the wrong process when no clear match exists. While this can be adjusted in the prompt, the model might still return incorrect results without indicating uncertainty. We also observed that the LLMs sometimes tend to hallucinate, by slightly changing the names. In addition, we observed result variability both across repeated runs of the same LLM and between different LLM implementations during testing. In some cases, the LLM failed to achieve sufficient mappings in one run despite succeeding in another. This variability raises significant concerns regarding reproducibility. To address this, coupling with or validation through complementary approaches, such as fuzzy logic methods demonstrated in other fields of research [19,20], should be considered. Unit conversion, while generally successful, revealed inconsistencies in specific cases. The framework's current implementation applies conversions without providing transparency about the assumptions or sources for density values and other conversion factors.

While our evaluation exclusively used open-weight LLMs, this choice may have limited performance compared to commercially available models with potentially superior capabilities. Since only Alibaba Cloud Qwen 3 235B A22B Thinking 2507 successfully handled our task, and it also performs better in general benchmarks, compared to the other LLMs we used [47], this may indicate that higher overall model performance is beneficial for LCA applications. It also shows that the task involves a level of complexity that exceeds the capabilities of basic LLMs. The runtimes recorded for the LCI mapping may not be fully representative, as they are influenced by the performance of the hosting platform and its system load.

4.4. Methodological considerations

Our current implementation lacks testing across diverse LCI datasets, for example, representing different industries and data types, as well as database and LLM switches, all of which could lead to substantially different outputs. In the prompt, we did not allow the model to output "unknown" for uncertain mapping. This design choice, while ensuring mapping completeness, might have sacrificed precision. However, allowing "unknown" entries would result in unmapped processes, raising the question of whether the LCA practitioner must intervene at this stage to model the missing processes or whether LLMs can autonomously handle such gaps, for instance, by extracting LCI data from literature. This issue requires further testing and discussion, particularly regarding the desired level of practitioner involvement and the extent to which LLMs can effectively resolve such data gaps. Furthermore, the prompt itself could be further evaluated using alternative approaches, such as few-shot learning [48].

A critical methodological gap is the absence of confidence metrics or iterative refinement mechanisms. Without such features, users cannot differentiate between high-confidence and potentially problematic mappings that require expert review. However, such validation effort, most probably required due to the opaque, "black-box" nature of LLMs, may approach or even exceed the time investment of manual modeling.

Furthermore, using a general-purpose LLM for this task may represent an inefficient use of computational resources. A rule-based system might be more resource-efficient but lacks flexibility regarding database updates, user-specific requirements, and dynamic factors. Alternatively, approaches such as retrieval-augmented generation and fine-tuning could be applied to customize a general-purpose model and potentially allow the use of smaller models that, with these techniques, are still capable of performing the mapping task.

The simplified heuristic approach for distinguishing processes and elementary flows, as well as handling by-products, does not align with the full spectrum of LCA methods and may result in inconsistencies and non-preferred methodological choices. At present, Brightway 2 is used, and expanding framework compatibility to the latest Brightway 2.5 version is necessary.

5. Impact

FAULDIER has the potential to open new research directions by questioning whether LCA modeling can be accelerated through the direct use of heterogeneous LCI data, thereby reducing or even eliminating the need for manual curation. This raises an important question about whether such automation is feasible given the inherent inconsistencies and sometimes non-reproducible nature of LLMs, which could compromise the precision required in LCA studies.

Another aspect to examine is whether this approach could broaden the applicability of LCA by dynamically adapting to data derived from the outputs of other software tools. If successful, this capability might lead to an increase in the number of environmental assessments conducted, ultimately contributing to improved environmental performance overall by simplifying and accelerating the LCA process.

Beyond enabling new research questions, FAULDIER also enhances the pursuit of existing ones. By supporting the integration of outputs from other tools, it facilitates software coupling, which is particularly valuable for method integration. In the context of prospective LCA, where multiple database modifications are often employed to represent scenarios or uncertainty ranges, FAULDIER could significantly ease implementation.

Although some features remain under development, the framework could have the potential to transform daily practice for LCA practitioners by reducing the need for manual curation. Its ability to potentially adapt to database updates, even when naming conventions or process structures change, further reduces the effort required to maintain models over time. In the future, these advancements could automate workflows to a greater extent, saving time, minimizing human error, and ultimately accelerating sustainability decision-making support. However, it remains to be investigated to what extent such automation is fully achievable with this type of system.

6. Conclusions and future work

While mapping accuracy, current technical, and licensing constraints limit FAULDIER's immediate applicability to LCA workflows, future advancements in both LLM capabilities and database accessibility, combined with further methodological implementations within FAULDIER, could potentially support LLM-assisted LCA workflows, provided that the reproducibility of generated results can be ensured. To validate this potential, further testing is required across various LCI datasets and databases, input formats, as well as improved data-handling and prompting strategies. Achieving this will require addressing both technical and licensing challenges.

Future work should include testing FAULDIER on previously modeled LCAs. This would allow a direct comparison between manual and automated results and help identify key drivers for deviations, such as poorly mapped processes and elementary flows. Multiple runs would need to be performed on this dataset to account for potential variations in the LLM. To reduce the trade-off between automation and mapping success, implementing mechanisms to quantify mapping confidence and provide alternative mappings would enhance transparency and reduce validation effort. An iterative procedure of feeding back low-confidence mappings to another LLM run could improve the mapping. Alternatively, the development of fine-tuned LLMs on LCA datasets could improve mapping precision and alignment with LCA practitioner preferences. Moreover, mapping performance could also be enhanced by ensemble approaches, where multiple LLMs are queried and their results cross-validated, or even through models "talking to each other" during the mapping process to reach consensus. With these implementations, the question of whether LLMs can be integrated into LCA workflows may be addressed more comprehensively.

CRediT authorship contribution statement

Lukas Lazar: Writing – review & editing, Writing – original draft, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work used the Scientific Compute Cluster at GWDG, the joint data center of Max Planck Society for the Advancement of Science (MPG) and University of Göttingen. The authors gratefully acknowledge the computing time granted by the KISSKI project and the access to

SAIA, which provided the API access to Large Language Models for this study.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.softx.2026.102602.

References

[1] Guinée JB, Heijungs R, Huppes G, Zamagni A, Masoni P, Buonamici R, et al. Life cycle assessment: past, present, and future. Env Sci Technol 2011;45:90–6. https://doi.org/10.1021/es101316v.
[2] Amahmoud A, El Attar MM, Meleishy A. The evolution of life cycle assessment approach: a review of past and future prospects. IOP Conf Ser: Earth Env Sci 2022; 992:012002. https://doi.org/10.1088/1755-1315/992/1/012002.
[3] Curran MA. Life-cycle assessment: principles and practice. National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency; 2006.
[4] Hauschild MZ, Rosenbaum RK, Olsen SI. Life cycle assessment: theory and practice. editors. Cham: Springer International Publishing; 2018. https://doi.org/10.1007/978-3-319-56475-3.
[5] Cays J. An environmental life cycle approach to design: LCA for designers and the design market. Cham: Springer International Publishing; 2021. https://doi.org/10.1007/978-3-030-63802-3.
[6] Aspray W. Computing before computers. Ames: Iowa State University Press; 1990. p. 50010.
[7] PRé Sustainability. 35 years of SimaPro and the question that started it all. SimaPro 2026. https://simapro.com/insights/35-years-of-simapro-and-the-question-that-started-it-all/ (accessed February 5, 2026).
[8] Mutel C. Brightway: An open source framework for Life cycle assessment. JOSS 2017;2:236. https://doi.org/10.21105/joss.00236.
[9] Jolivet R, Clavreul J, Brière R, Besseau R, Prieur Vernat A, Sauze M, et al. lca_algebraic: a library bringing symbolic calculus to LCA for comprehensive sensitivity analysis. Int J Life Cycle Assess 2021;26:2457–71. https://doi.org/10.1007/s11367-021-01993-z.
[10] Sacchi R, Terlouw T, Siala K, Dirnaichner A, Bauer C, Cox B, et al. PRospective EnvironMental Impact asSEment (premise): a streamlined approach to producing databases for prospective life cycle assessment using Integrated Assessment Models. Renew Sustain Energy Rev 2022;160:112311. https://doi.org/10.1016/j.rser.2022.112311.
[11] Volkart K, Mutel CL, Panos E. Integrating life cycle assessment and energy system modelling: methodology and application to the world energy scenarios. Sustain Prod Consum 2018;16:121–33. https://doi.org/10.1016/j.spc.2018.07.001.
[12] Umeda Y, Fukushige S, Kunii E, Matsuyama Y. LC-CAD: a CAD system for life cycle design. CIRP Ann 2012;61:175–8. https://doi.org/10.1016/j.cirp.2012.03.043.
[13] Rijcken E, Zervanou K, Mosteiro P, Scheepers F, Spruit M, Kaymak U. Machine learning vs. rule-based methods for document classification of electronic health records within mental health care—A systematic literature review. Nat Lang Process J 2025;10:100129. https://doi.org/10.1016/j.nlp.2025.100129.
[14] Kautz HA. The third AI summer: AAAI Robert S. Engelmore Memorial Lecture. AI Mag 2022;43:105–25. https://doi.org/10.1002/aaai.12036.
[15] Yaguinuma CA, Afonso GF, Ferraz V, Borges S, Santos MTP. A fuzzy ontology-based semantic data integration system. J Inform Knowl Manage 2012;10:285–99. https://doi.org/10.1142/S0219649211002997.
[16] Shoaip N, El-Sappagh S, Abuhmed T, Elmogy M. A dynamic fuzzy rule-based inference system using fuzzy inference with semantic reasoning. Sci Rep 2024;14:4275. https://doi.org/10.1038/s41598-024-54065-1.
[17] González B., Adenso-Díaz B., G. PL. A fuzzy logic approach for the impact assessment in LCA. Resources, conservation and recycling 2002;37:61–79. https://doi.org/10.1016/S0921-3449(02)00069-1.
[18] Tung WL, Quek C. eFSM—A novel online neural-fuzzy semantic memory model. IEEe Trans Neural Netw 2010;21:136–57. https://doi.org/10.1109/TNN.2009.2035116.
[19] Zhang K, Tsang YP, Lee CKM, Wu CH. Integrating large language models with explainable fuzzy inference systems for trusty steel defect detection. Pattern Recognit Lett 2025;192:29–35. https://doi.org/10.1016/j.patrec.2025.03.017.
[20] Figueiredo V. A fuzzy logic prompting framework for large language models in adaptive and uncertain tasks. arXiv preprint 2025. https://arxiv.org/html/2508.06754v1.
[21] Ballon M, Algaba A, Ginis V. The relationship between reasoning and performance in large language models – o3 (mini) thinks harder. Not Long arXiv preprint 2025. https://doi.org/10.48550/arXiv.2502.15631.
[22] Preuss N, Alshehri AS, You F. Large language models for life cycle assessments: opportunities, challenges, and risks. J Clean Prod 2024;466:142824. https://doi.org/10.1016/j.jclepro.2024.142824.
[23] Kumar A, Nazemi F, Kodamana H, Ramteke M, Bakshi BR. A large language model-based framework to retrieve life cycle inventory and environmental impact data from scientific literature. Env Sci Technol 2025;59:22533–43. https://doi.org/10.1021/acs.est.5c05955.
[24] Sitaraman A, Balaji B, Agarwal Y. SpiderGen: towards procedure generation for carbon life cycle assessments with generative AI. arXiv preprint 2025. https://doi.org/10.48550/arXiv.2511.10684.
[25] Balaji B, Ebrahimi F, G Domingo NG, Vunnava VSG, Faridee A-Z, Ramalingam S, et al. Emission factor recommendation for life cycle assessments with generative AI. Env Sci Technol 2025;59:9113–22. https://doi.org/10.1021/acs.est.4c12667.
[26] Zhang X, Guo X, Zhao J, Xiong J, Tian Y. Intelligent application of large language model to life cycle assessment methodology. J Clean Prod 2025;529:146776. https://doi.org/10.1016/j.jclepro.2025.146776.
[27] Donaldson A, Balaji B, Oriekezie C, Kumar M. Patouillard L., An expert-grounded benchmark of general purpose LLMs in LCA. arXiv preprint 2025. https://doi.org/10.48550/arXiv.2510.19886.
[28] Preuss N, You F. Automating life cycle assessments through artificial intelligence agents and integrated assessment models. Env Sci Technol 2026;60:33–48. https://doi.org/10.1021/acs.est.5c14493.
[29] Tu Q, Guo J, Li N, Qi J, Xu M. Mitigating grand challenges in life cycle inventory modeling through the applications of large language models. Env Sci Technol 2024;58:19595–603. https://doi.org/10.1021/acs.est.4c07634.
[30] Brightway Developers. Installation — Brightway documentation 2025. https://docs.brightway.dev/en/legacy/content/installation/installation.html (accessed August 19, 2025).
[31] Wernet G, Bauer C, Steubing B, Reinhard J, Moreno-Ruiz E, Weidema B. The ecoinvent database version 3 (part I): overview and methodology. Int J Life Cycle Assess 2016;21:1218–30. https://doi.org/10.1007/s11367-016-1087-8.
[32] Schmidt JH. Documentation of the Data consolidation and calibration exercise, and the scenario parameterisation. Overall Mapp Phys Flows Stocks Resour Forecast Waste Quant Eur Identify Life-Cycle Environ Stakes Waste Prev Recycl Deliv 2010. 6–1.
[33] ecoinvent Association. End User Licence Agreement (EULA) version 3 of 1 April 2022. Zürich, Switzerland: 2022. https://ecoinvent.org/app/uploads/2024/07/EULA_new_branding_08_11_2023.pdf.
[34] Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen. Available models. Documentation for HPC n.d. https://docs.hpc.gwdg.de/services/chat-ai/models/index.html (accessed August 21, 2025).
[35] Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen. SAIA. Documentation for HPC n.d. https://docs.hpc.gwdg.de/services/saia/index.html (accessed August 21, 2025).
[36] Doosthosseini A, Decker J, Nolte H, Kunkel JM. Chat AI: a seamless slurm-native solution for HPC-based services. arXiv preprint 2024. https://doi.org/10.48550/arXiv.2407.00110.
[37] Lazar L. Prospective life cycle assessment of early-stage CO_2-removing artificial photosynthesis producing synthesis gas. Hydrog Perox Meth Energ Technol 2025; 13:2402224. https://doi.org/10.1002/ente.202402224.
[38] Hwang Y, Kim Y, Koo J, Kang T, Bae H, Jung K. LLMs can be easily confused by instructional distractions. editors. In: Che W, Nabende J, Shutova E, Pilehvar MT, editors. Proceedings of the 63rd annual meeting of the association for computational linguistics. 1. Vienna, Austria: Association for Computational Linguistics; 2025. p. 19483–96. https://doi.org/10.18653/v1/2025.acl-long.957.
[39] Kalai AT, Nachum O, Vempala SS, Zhang E. Why language models hallucinate. arXiv preprint 2025. https://doi.org/10.48550/arXiv.2509.04664.
[40] Balaji B, Vunnava VSG, Domingo N, Gupta S, Gupta H, Guest G, et al. Flamingo: environmental impact factor matching for life cycle assessment with zero-shot machine learning. ACM J Comput Sustain Soc 2023;1:1–23. https://doi.org/10.1145/3616385.
[41] Makersite. The AItightrope in sustainability: balancing automation, accuracy, and trust in LCA/EPD. https://pages.makersite.io/hubfs/Makersite_AI_LCA_Trust_Whitepaper_2025.pdf; 2025.
[42] Zhang Z, Thavikulwat P, Metzger AL, Mei Y, Hähnlein F, Englhardt Z, et al. Living sustainability: in-context interactive environmental impact communication. Proc ACM Interact Mob Wear Ubiquit Technol 2025;9:1–42. https://doi.org/10.1145/3749488.
[43] Teng Y, Li CZ, Shen GQP, Yang Q, Peng Z. The impact of life cycle assessment database selection on embodied carbon estimation of buildings. Build Env 2023; 243:110648. https://doi.org/10.1016/j.buildenv.2023.110648.
[44] Jiang H, Wu Q, Lin C-Y, Yang Y, LLMLingua QL. Compressing prompts for accelerated inference of large language models. In: Proceedings of the 2023 conference on empirical methods in natural language processing. Singapore: Association for Computational Linguistics; 2023. p. 13358–76. https://doi.org/10.18653/v1/2023.emnlp-main.825.
[45] Codina V, Ricci F, Ceccaroni L. Semantically-enhanced pre-filtering for context-aware recommender systems. In: Proceedings of the 3rd workshop on context-awareness in retrieval and recommendation. Rome Italy: ACM; 2013. p. 15–8. https://doi.org/10.1145/2442670.2442674.
[46] Wright MM, Tan ECD, Tu Q, Martins A, Parvatker AG, Yao Y, et al. Life cycle inventory availability: status and prospects for leveraging new technologies. ACS Sustain Chem Eng 2024;12:12708–18. https://doi.org/10.1021/acssuschemeng.4c02519.
[47] Hugging Face. Open LLM leaderboard - a hugging face space by open-llm-leaderboard 2024. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard (accessed December 5, 2025).
[48] Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language models are few-shot learners. Adv Neural Inf Process Syst 2020:33. https://doi.org/10.48550/arXiv.2005.14165.