PALIMPSEST: BILL OF MATERIALS PREDICTION - A CASE STUDY WITH SOLID STATE DRIVES

Anran Wang, Zaid Thanawala, Harsh Gupta, Jared Kramer, Kommy Weldemariam , Bharathan Balaji [*]
Amazon, USA
{anrwang, ztthanaw, hrshgup, jaredkra, kommy, bhabalaj}@amazon.com

ABSTRACT

Accurately quantifying product carbon footprints (PCFs) is critical for organizations to measure environmental impacts and develop decarbonization strategies. However, traditional methods require Bills of Materials (BOMs) data as a key input for PCF estimation, which is time-intensive and limits scalability. We present Palimpsest, an automated BOM generation algorithm given product specification as input using Large Language Models (LLMs) and a reference dataset. Palimpsest extracts data from teardown reports to build a BOM repository, retrieves reference products based on an their attribute list, generates BOMs by systematically modifying reference BOMs based on attribute differences, and standardizes the output to enable automated PCF estimation. We also introduce a novel impact-based evaluation framework that compares predicted BOMs with ground truth, focusing on the accuracy in carbon impact. We benchmark our model against a naive LLM solution and a traditional PCF estimation approach for solid state drives and find it outperforms these methods with a weighted F1 of 99.5%. By streamlining and automating BOM prediction, our method reduces the manual effort required for PCF estimation, driving progress toward net-zero emissions targets across industries.

1 INTRODUCTION AND MOTIVATION

The emission of greenhouse gases (GHGs), particularly carbon dioxide (CO_2), is the primary driver of global climate change (Change et al., 2014). Organizations worldwide have pledged to achieve net-zero carbon emissions and publicly release their annual GHG emissions as part of their sustainability reports (The Climate Pledge, 2024; Black et al., 2021). Accurately quantifying the GHG emissions associated with products and activities is crucial for organizations to understand and mitigate their environmental impact. The International Organization for Standardization (ISO) introduced the ISO 14067:2018 standard, which provides comprehensive principles, requirements, and guidelines for calculating a product carbon footprint (PCF). This standard enables practitioners to assess and report emissions across a product's lifecycle, from raw material extraction to disposal.

Despite the availability of such standards, organizations face significant challenges in implementing PCF assessments (Balaji et al., 2023a). Challenges include the high costs associated with data collection, substantial time investments, and the lack of modeling consensus (Beattie, 2012). We focus on the collection of the Bill of Materials (BOM) — a comprehensive list of all parts, materials, and quantities required to manufacture a product. The BOM is essential to estimate the carbon impacts of materials and manufacturing processes (Babbitt et al., 2020), which account for ∼40% of emissions (Meinrenken et al., 2020). Traditional methods of compiling BOM data are time-intensive and resource-heavy, often requiring subject matter experts to invest weeks or even months to collect and validate the necessary information (Bicalho et al., 2017; Miah et al., 2018). This bottleneck limits the scalability of PCF assessments, particularly for organizations with diverse product portfolios. To address the challenge of data availability, previous efforts have employed statistical methods to estimate missing data (Moreau et al., 2012; Zargar et al., 2022). While these methods have shown

---

promise, they typically require practitioners to identify potential missing materials or components manually, limiting their scalability and generalizability.

Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) systems have been applied to carbon measurement, supporting tasks such as emission factor mapping (Balaji et al., 2024) and carbon emission reduction planning (Gopalakrishnan et al., 2024). LLMs have been utilized as virtual subject matter experts (SMEs), offering recommendations for carbon accounting, and evaluated against human experts Luo et al. (2023). However, the application of LLMs in generation tasks that require a deep understanding of physics, mechanics, and materials remains largely under-explored (Chang et al., 2024). This gap is particularly significant in areas such as BOM generation, where the synthesis of domain-specific knowledge with structured reasoning is critical. Addressing this limitation opens new possibilities for leveraging LLMs in highly technical and knowledge-intensive domains, including sustainable product design and decarbonization planning.

Additionally, existing RAG evaluation frameworks have focused on comparing generated text to external knowledge used during retrieval (Yu et al., 2024; Ru et al., 2024). While this approach evaluates factual consistency, it falls short in assessing tasks such as BOM prediction where the output is a list of items with an importance weight attached to each item. The output needs to be machine-readable so that the total carbon impact of the predicted BOM can be computed and evaluated with metrics such as mean absolute percentage error (MAPE). To ensure the critical items were correctly identified, the text in each line item needs to be standardized to a pre-defined vocabulary, and the units of measure needs to be normalized. Similar evaluation methods is required for tasks such as food ingredient prediction (Li et al., 2019) and energy disaggregation (Batra et al., 2014).

Contributions: To our knowledge, we are the first to explore methods for BOM prediction and evaluation. Our algorithm, called Palimpsest, combines data extraction, information retrieval, and structured text generation with LLM. Palimpsest generates output in a standardized output that we evaluate using metrics such as MAPE and weighted F1 score. Palimpsest requires minimal input of products, provides human-readable justifications to enable expert verification, and substantially reduces the manual effort required for PCF assessments.

2   METHODS

Given a product and its attributes, our objective is to predict the product's BOM. We use the predicted BOM to compute the PCF and identify the highest emission causing items, called 'hotspots', that shows where to focus carbon reduction investments. We picked solid state drive (SSD) for our case study as it is a representative electronics product, and contributes >75% to the manufacturing emissions of a compute server (Tannu & Nair, 2023). An SSD has 25+ components that are not evident from product specifications, companies request the BOM from the manufacturer or rely on a teardown. A predicted BOM leapfrogs these steps and accelerates decision making.

The end-to-end process of Palimpsest encompasses four components: extraction, retrieval, generation, and standardization, as illustrated in Figure 1. We use Claude 3.5 Sonnet as the LLM (Anthropic, 2025), and our prompts (listed in Appendix D) range from zero-shot to few-shot.

Extraction - We create a BOM repository by extracting data from teardown documents and organize it into a multilevel schema that captures materials and associated manufacturing processes (Appendix C). We developed an LLM-powered extraction method to handle diverse file types (spreadsheets and PDFs), parse multiple tables within a single document, and normalize heterogeneous table structures. The extraction process produces two outputs: (1) a BOM repository and (2) a product attribute library. For solid state drives (SSDs), the attributes include storage capacity, RAM type, product weight, and release date.

Retrieval - When a product is presented, Palimpsest identifies a reference product and its associated BOM from the product attribute library and BOM repository. This step employs a few-shot LLM to determine the most similar product within the repository. We rank the list of attributes to highlight the relative importance of each attribute in determining product similarity. By prioritizing fundamental characteristics — such as (1) storage capacity, (2) RAM, (3) product weight, and (4) release date — the ranked list enables the LLM to focus on the most significant factors, ensuring robust and contextually relevant retrieval.
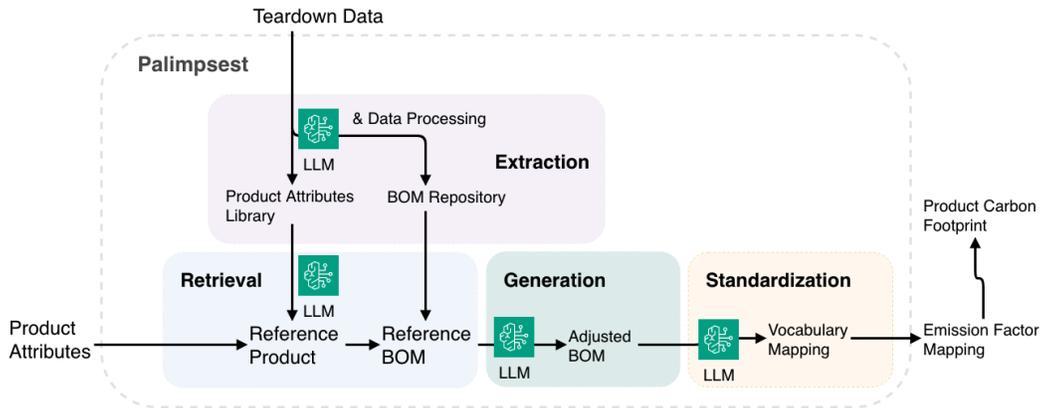
Figure 1: Palimpsest consists of four components - extraction, retrieval, generation, and standardization. Each model generated output contains justifications for its choices or modifications made.

Generation - Once a reference product is identified, Palimpsest leverages its BOM as a baseline to generate a BOM for the given product. This step involves instructing the LLM to systematically modify the reference BOM to account for attribute differences between the reference and the given product. Adjustments include changing mass or quantity, adding or removing components, updating manufacturing processes, and revising component descriptions to reflect variations in component specifications, materials, and other relevant factors inferred from the attribute discrepancies. A real example of Retrieval and Generation can be found in Appendix B.

Standardization - LLM-generated data often include free-form text, making it challenging for practitioners to compare BOMs when different terminologies are used interchangeably. For example, "integrated circuits" can be described with terms such as "ICs" or "chips". To address this issue, the final step of Palimpsest is a standardization process. This step maps the generated material types and manufacturing processes to a predefined vocabulary, ensuring consistency and enabling seamless comparison and analysis across BOMs.

PCF Generation - We estimate the carbon impact of each line item of the predicted BOM by mapping it to an *emission factor* using an LLM based algorithm adapted from Balaji et al. (2024). An emission factor provides the estimated greenhouse gas emissions associated with unit quantity of a material or process measured in kilograms of carbon dioxide equivalent ($kgCO_2e$), e.g. the emission factor for 'pear production' is 0.31 $kgCO_2e$ per kg of pear (Wernet et al., 2016). The unit $kgCO_2e$ accounts for emissions from multiple greenhouse gases, such as methane and nitrous oxide, and normalizes the global warming potential to that of carbon dioxide (Kikstra et al., 2022). We use emission factor datasets from Ecoinvent (Wernet et al., 2016) and Sphera (2025). We compute the total carbon footprint as the addition of individual impacts following methods used by Amazon Devices (2024). As we are interested in evaluating the predicted BOM, we only estimate the emissions from the materials and manufacturing phase, and ignore impact of use and disposal.

3 RESULTS

We evaluate the performance of Palimpsest with 112 teardown reports of SSDs spanning 2012 to 2024, sourced from TechInsights Inc.. Each report contains product design, components, and manufacturing processes, and are all processed through the Extraction step. In the Retrieval step, a leave-one-out approach is used for evaluation. Specifically, for each SSD, its attributes are excluded from the product attribute library when searching for a reference product.

The performance of Palimpsest is benchmarked against a Naive LLM approach, in which the model is tasked to generate a BOM from scratch using the same product attributes employed in Palimpsest, we include the prompts in Appendix E. The Naive LLM method relies on the world knowledge used during training, similar to the method used by Luo et al. (2023). The output does not include reference sources that an expert can use to audit to the results. We also evaluate with Environmentally-

Table 1: Performance of Palimpsest against Naive LLM BOM prediction and EEIO model.

Model	MAPE	RMSE	Weighted F1
Palimpsest	64.64%	**211.80**	**0.9948**
Naive LLM	72.43%	264.76	0.9601
EEIO	**62.24%**	239.28	NA

Extended Input-Output (EEIO) model, an industry standard widely adopted for calculating carbon emissions (Balaji et al., 2023b). This is a top-down macroeconomic approach using cost data to estimate the emissions at an industry sector level, the method only outputs the final carbon footprint and does not provide a breakdown of emissions.

We use Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) for evaluating the performance of BOM prediction. MAPE represents the average of absolute percentage errors between calculated carbon emission from groundtruth and predicted BOM, while RMSE quantifies the magnitude of prediction errors, emphasizing larger deviations between the predicted and ground truth carbon emissions.

To evaluate the model's ability to generate a complete BOM structure with relevant materials and components, we use weighted $F1$ score (equation in Appendix F.2). This metric is adjusted based on the carbon impact associated with each material, giving higher weight to components with a larger environmental footprint. The F1 score is a harmonic mean of precision and recall, where precision measures whether each of the predicted items is correct and recall measures whether all the ground truth items were predicted. Unlike MAPE and RMSE, weighted F1 focuses on whether the model correctly identifies high impact BOM items, rather than accurately predicting their quantities. This ensures that critical, high-impact materials are prioritized in the evaluation. EEIO does not have this metric because it doesn't generate BOMs.

Table 1 presents a comparison of validation performance between Palimpsest and the baselines. Palimpsest outperforms the Naive LLM approach in all metrics, highlighting the benefits of leveraging a reference-based retrieval mechanism in BOM prediction. One key observation is a lack of diversity in the BOMs predicted by Naive LLM, which consistently underestimates carbon emissions. Of the 112 predicted BOMs, only one has emissions exceeding 150 kgCO$_2$e, compared to 42 BOMs in the ground truth within that category, with some emissions as high as 927 kgCO$_2$e. In contrast, Palimpsest improves this distribution significantly, with 30 predicted BOMs exceeding 150 kgCO$_2$e. Appendix F.3 provides a full distribution comparison for predicted and ground truth emissions of both models.

Our method achieves parity with EEIO estimates in terms of MAPE, while outperforming in RMSE. Unlike EEIO, which is limited to estimating total carbon emissions, Palimpsest generates a comprehensive BOM. This capability enables hotspot analysis, offering a more detailed and actionable foundation for environmental assessments and facilitating targeted strategies for emission reduction.

4 LIMITATIONS AND FUTURE WORK

Palimpsest can predict BOMs within minutes, saving both time and costs associated with PCF. Practitioners can use this tool as a jump start for hotspot analysis, supporting organizations decarbonization strategy planning. However, current work is limited to one product type, and future work is needed to evaluate the generalizability and scalability of this model across products and domains, especially in data-sparse areas. Our current approach relies on an expert-defined ranked list of product attributes, which is product-specific and may become a bottleneck as the model is generalized. Automating this ranking process is a critical area for improvement to enhance the scalability and adaptability of the framework. BOM verification is another challenge. While Palimpsest predictions are evaluated using metrics like MAPE and RMSE, practitioners often require BOM completeness and consistency for their use cases. Each LLM-generated BOM line item needs to be individually validated to meet these requirements. Developing automated verification techniques will be essential to ensure the reliability of the generated BOMs.

REFERENCES

Amazon Devices. Amazon Devices Product Carbon Footprint Methodology. `https://sustainability.aboutamazon.com/devices-carbon-methodology.pdf`, 2024. Accessed: 2025-02-01.

Anthropic. Claude 3.5 Sonnet. `https://www.anthropic.com/claude/sonnet`, 2025. Accessed: 2025-02-01.

Callie W Babbitt, Hema Madaka, Shahana Althaf, Barbara Kasulaitis, and Erinn G Ryen. Disassembly-based bill of materials data for consumer electronic products. *Scientific Data*, 7 (1):251, 2020.

Bharathan Balaji, Geoffrey Guest, Venkata Sai Gargeya Vunnava, Jared Kramer, Aravind Srinivasan, and Michael Taptich. Scaling carbon footprinting: Challenges and opportunities. In *Proceedings of the AAAI Symposium Series*, volume 2, pp. 35–39, 2023a.

Bharathan Balaji, Venkata Sai Gargeya Vunnava, Geoffrey Guest, and Jared Kramer. Caml: Carbon footprinting of household products with zero-shot semantic text similarity. In *Proceedings of the ACM Web Conference 2023*, pp. 4004–4014, 2023b.

Bharathan Balaji, Fahimeh Ebrahimi, Nina Domingo, Gargeya Vunnava, Abu-Zaher Faridee, Soma Ramalingam, Shikha Gupta, Anran Wang, Harsh Gupta, Domenic Belcastro, Kellen Axten, Jeremie Hakian, Jared Kramer, Aravind Srinivasan, and Qingshi Tu. Parakeet: Emission factor recommendation for carbon footprinting with generative ai. 2024. URL `https://www.climatechange.ai/papers/neurips2024/52`.

Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. Nilmtk: An open source toolkit for non-intrusive load monitoring. In *Proceedings of the 5th international conference on Future energy systems*, pp. 265–276, 2014.

Geoffrey Beattie. Psychological effectiveness of carbon labelling. *Nature Climate Change*, 2:214–217, 04 2012. doi: 10.1038/nclimate1468.

Tereza Bicalho, Ildo Sauer, Alexandre Rambaud, and Yulia Altukhova. Lca data quality: A management science perspective. *Journal of Cleaner Production*, 156:888–898, 2017.

Richard Black, Kate Cullen, Byron Fay, Thomas Hale, John Lang, Saba Mahmood, and Steve Smith. Taking stock: A global assessment of net zero targets. *Energy & Climate Intelligence Unit and Oxford Net Zero*, 23, 2021.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

IPCC Climate Change et al. Mitigation of climate change. *Contribution of working group III to the fifth assessment report of the intergovernmental panel on climate change*, 1454:147, 2014.

Varsha Gopalakrishnan, Shaena Ulissi, Andrew Dumit, Krishna Rao, Katherine Tsai, and Sangwon Suh. How are companies reducing emissions? an llm-based approach to creating a carbon emissions reduction levers library at scale. In *NeurIPS 2024 Workshop on Tackling Climate Change with Machine Learning*, 2024. URL `https://www.climatechange.ai/papers/neurips2024/90`.

ISO 14067:2018. Greenhouse gases — Carbon footprint of products — Requirements and guidelines for quantification. Standard, International Organization for Standardization, August 2018.

Jarmo S Kikstra, Zebedee RJ Nicholls, Christopher J Smith, Jared Lewis, Robin D Lamboll, Edward Byers, Marit Sandstad, Malte Meinshausen, Matthew J Gidden, Joeri Rogelj, et al. The ipcc sixth assessment report wgiii climate assessment of mitigation pathways: from emissions to global temperatures. *Geoscientific Model Development*, 15(24):9075–9109, 2022.

Jiatong Li, Ricardo Guerrero, and Vladimir Pavlovic. Deep cooking: Predicting relative food ingredient amounts from images. In *Proceedings of the 5th international workshop on multimedia assisted dietary management*, pp. 2–6, 2019.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. Interpretable long-form legal question answering with retrieval-augmented large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 22266–22275, 2024.

Biao Luo, Jinjie Liu, Zhu Deng, Can Yuan, Qingrun Yang, Lei Xiao, Yucong Xie, Fanke Zhou, Wenwen Zhou, and Zhu Liu. Autopcf: A novel automatic product carbon footprint estimation framework based on large language models. In *Proceedings of the AAAI Symposium Series*, volume 2, pp. 102–106, 2023.

Christoph J Meinrenken, Daniel Chen, Ricardo A Esparza, Venkat Iyer, Sally P Paridis, Aruna Prasad, and Erika Whillas. Carbon emissions embodied in product value chains and the role of life cycle assessment in curbing them. *Scientific Reports*, 10(1):6184, 2020.

Jamal Hussain Miah, Andrew Griffiths, Ryan McNeill, Sharla Halvorson, Urs Schenker, Namy Espinoza-Orias, Stephen Morse, Aidong Yang, and Jhuma Sadhukhan. A framework for increasing the availability of life cycle inventory data based on the role of multinational companies. *The international journal of life cycle assessment*, 23:1744–1760, 2018.

Vincent Moreau, Gontran Bage, Denis Marcotte, and Réjean Samson. Statistical estimation of missing data in life cycle inventory: an application to hydroelectric power plants. *Journal of Cleaner Production*, 37:335–341, 2012.

Zabir Al Nazi and Wei Peng. Large language models in healthcare and medical domain: A review. In *Informatics*, volume 11, pp. 57. MDPI, 2024.

Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, et al. Ragchecker: A fine-grained framework for diagnosing retrieval-augmented generation. *arXiv preprint arXiv:2408.08067*, 2024.

Sphera. Product Life Cycle Assessment Solutions. 2025.

Swamit Tannu and Prashant J Nair. The dirty secret of ssds: Embodied carbon. *ACM SIGENERGY Energy Informatics Review*, 3(3):4–9, 2023.

TechInsights Inc. Techinsights inc. - the semiconductor information platform. `https://www.techinsights.com/`. Accessed: 2025-01-06.

The Climate Pledge. Be the planet's turning point: Join the world's top companies—and take action now to reach net-zero carbon by 2040. `https://www.theclimatepledge.com/us/en`, 2024.

Gregor Wernet, Christian Bauer, Bernhard Steubing, Jürgen Reinhard, Emilia Moreno-Ruiz, and Bo Weidema. The ecoinvent database version 3 (part i): overview and methodology. *The International Journal of Life Cycle Assessment*, 21:1218–1230, 2016.

Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. Evaluation of retrieval-augmented generation: A survey. *arXiv preprint arXiv:2405.07437*, 2024.

Shiva Zargar, Yuan Yao, and Qingshi Tu. A review of inventory modeling methods for missing data in life cycle assessment. *Journal of Industrial Ecology*, 26(5):1676–1689, 2022.

Boyu Zhang, Hongyang Yang, Tianyu Zhou, Muhammad Ali Babar, and Xiao-Yang Liu. Enhancing financial sentiment analysis via retrieval augmented large language models. In *Proceedings of the fourth ACM international conference on AI in finance*, pp. 349–356, 2023.

A BACKGROUND AND RELATED WORK

Large Language Models (LLMs) have exhibited exceptional capabilities in generating coherent and contextually relevant text across various domains. To enhance their applicability and ensure precision, Retrieval-Augmented Generation (RAG) systems have gained prominence in critical fields such as legal, medical, and finance (Louis et al., 2024; Nazi & Peng, 2024; Zhang et al., 2023). By integrating external knowledge sources, RAG systems aim to augment the generative power of LLMs with accurate and contextually grounded information, addressing concerns around reliability and factuality in model outputs.

While existing RAG evaluation frameworks primarily focus on comparing generated text to the external knowledge used during the retrieval process (Yu et al., 2024; Ru et al., 2024), this narrow scope limits their applicability. Many RAG systems are designed not merely to retrieve and restate external information but to synthesize and infer new insights based on retrieved knowledge. These inferences, which often extend beyond simple information retrieval, play a pivotal role in practical applications where novel data generation is critical.

However, current evaluation methodologies fall short in assessing the accuracy, consistency, and plausibility of model-inferred information. This gap in evaluation frameworks poses significant challenges in validating the outputs of RAG systems, particularly when their utility hinges on generating accurate data as an SME. To bridge this gap, a comprehensive evaluation framework that accounts for both the retrieval fidelity and the validity of inferred content is essential for advancing the adoption and trustworthiness of RAG systems in real-world scenarios.

B PALIMPSEST WITH A REAL EXAMPLE

Figure 2 illustrates a real SSD example with data used and generated in each step.

C MULTI-LEVEL BOM SCHEMA

Table 2 shows the schema of the BOM repository.

D PALIMPSEST PROMPTS

D.1 PRODUCT ATTRIBUTES EXTRACTION PROMPT

```
### Instructions ###
You are a researcher at environmental protection agency. you want
to do a life cycle assessment(LCA) of bussiness activities based
on Bill of Material (BOM). You are interested in the environmental
impact associated with the materials and manufacturing phase of
the activity. You have given business activity descriptions, and
you want to extract the {product attributes} before building BOM.

Make the most of the given information. DO NOT say that
information is limited. Make sure you only extract from relevant
fields.
Double check whether there are more relevant information than the
one you chose.

Avoid filler words such as "Based on the details" or "happy to
assist", keep your response to the point.
Do not repeat the given instructions or information.
Return {product attributes} as a json list.

### Example Output Format ###
[{{
```
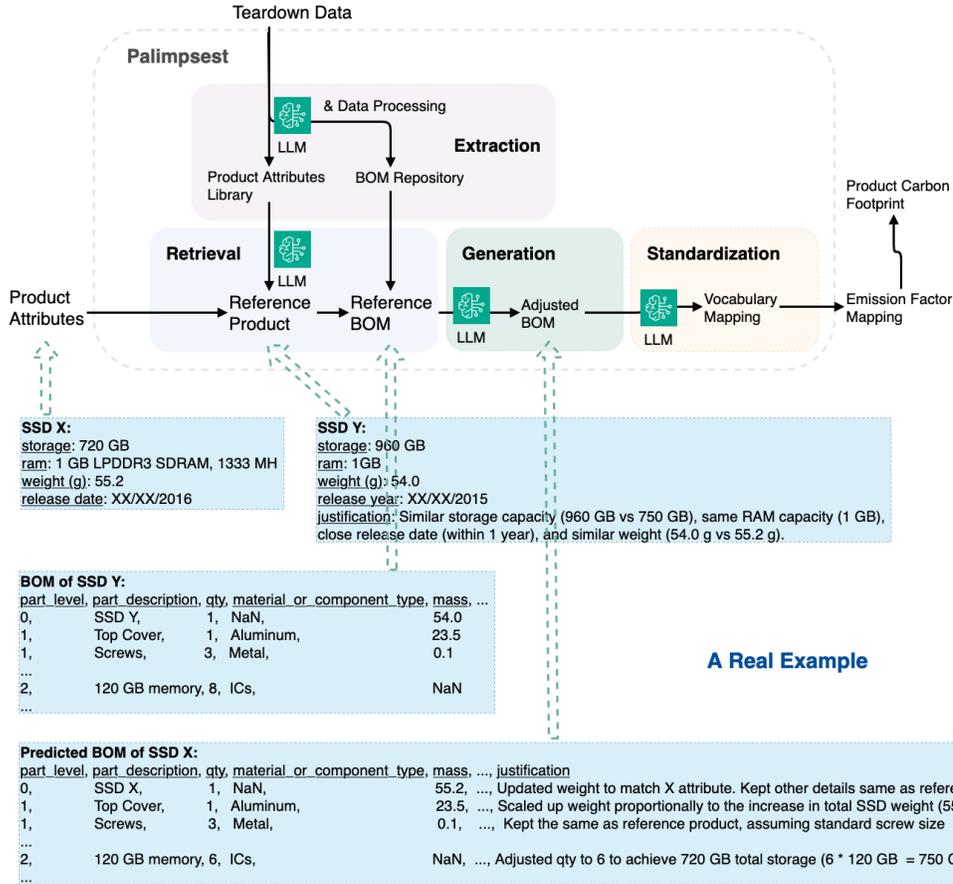
Figure 2: Palimpsest takes product attributes of SSD X as input and maps it to a reference product SSD Y in Retrieval. In Generation, it modifies the BOM of SSD Y and generates predicted BOM of SSD X.

```
  "activity_name": name of the activity or product,
  "primary_mfg_country": final assembly country (country of
  origin)
}}]

### Input and question ###
What is the {product attribute} of the BOM data given below:
Note: {attribute specific instruction}
{markdown data sheet}

Provide only the JSON list as the output, without any additional
text.
```

D.2 RETRIEVAL PROMPT

```
### Instructions ###
You are a researcher at an environmental protection agency
conducting a life cycle assessment (LCA) of products based on
their Bill of Materials (BOM). You are particularly interested in
calculating carbon footprint associated with the materials and
manufacturing phase of the product. You are given a TEST PRODUCT,
which is a {product type} with the following attributes:
```

Table 2: Multi-level BOM Schema for SSD

Name	Description
part_level	Part of tiered BOM structure. Top level system is 0 and each following sub-system/component is 1, 2, 3, etc.
part_type	A free text description of the material or component or part
part_description	A free text description of the material or component or part
qty	The amount part_description present in the product
material_or_component_type	Picked from a pre-defined list of available materials or components
material_percent_recycled_content	Required if recycled materials present (float range 0-100)
mass_g_per_unit	Mass of the part in grams
manufacturing_country	ISO2 Code
method_for_weight	Description of how the weight of the part was obtained
primary_manufacturing_process	Process selection from a pre-defined list
primary_mfg_yield_loss	Yield loss for the part. Part level yield loss refers to mass of wasted material divided by the mass of total input material which is the sum of wasted material and final desired material
ic_process_node	Process node of the integrated circuit from a pre-defined list.
ic_die_size_mm2	Die size of the integrated circuit based on supplier specification or lab x-ray. Die size indicates the functional area of the semiconducting material within individual integrated circuits package.
ic_package_type	Package type of the IC. Package indicates how semiconductors are assembled for final use. they vary in dimension (x-y-z), material, pin number, and bonding technology.
pcb_board_area_cm2	Area of the printed circuit board in product
pcb_type	Type of the PCB board from pre-defined options: Rigid or Flex
pcb_layers	Number of PCB or FPC layers

```
{product attributes in markdown}

You have access to a repository of products with available BOMs.
Your goal is to find the most similar products in the repository
as reference and use their BOMs to build a BOM for the TEST
PRODUCT. The reference products should provide an accurate basis
for estimating the materials, manufacturing processes, and overall
carbon footprint for the TEST PRODUCT.

### Key Considerations ###
1. **Environmental Impact**: Prioritize products with similar
environmental impacts, considering factors such as material
composition, weight, manufacturing date, and technology used
(e.g., type of NAND Flash, RAM technology).

2. **Technical Specifications**: Focus on products with similar
attributes. Here are the attributes you need to focus on together
with their importance rank. A lower number mean the attribute is
more important in determining whether two products are similar or
not.
```

```
{ranked attributes}

Additional key considerations:
{additional product specific rules}

### Task ###
Pick 5 reference products from the **product repository** and
rank them based on their similarity to the TEST PRODUCT in terms
of their attributes and environmental impact. Provide a detailed
justification for each choice, considering the factors listed
above. Ensure the products you choose have complete and relevant
data, and avoid choosing the TEST PRODUCT itself.

Make the most of the given information. Avoid filler words such
as "Based on the details" or "happy to assist", and keep your
response concise and to the point. Do not repeat the given
instructions or information.

### Product Repository ###
{product attributes library in markdown}

### Example Input ###
TEST PRODUCT: SSD4 with 1TB storage and 2GB DDR4 SDRAM,
official_release_date 1/1/2016, weight 45, uuid 100
Product repository:
    1. SSD1 with 2TB storage and 2GB DDR4 SDRAM,
    official_release_date 12/8/2016, weight 50, uuid 0
    2. SSD2 with 4TB storage and 2GB DDR4 SDRAM,
    official_release_date 2/1/2018, weight 71, uuid 1
    2. SSD3 with 480GB storage and 1GB DDR3 SDRAM,
    official_release_date 6/7/2014, weight 50, uuid 2

### Example Output ###
[{{
    'rank': 0
    'activity_name': 'SSD1',
    'official_release_date': 12/8/2016,
    'weight': 50,
    'ram_support': 2GB DDR4 SDRAM,
    'storage': 2TB,
    'uuid': 0,
    'justification': 'The most relevant product for SSD4 is SSD1
    because it has the same RAM techonogy and capacity.
    And storage is close to each other. This will make their
    carbon emission similar. '
}},
{{
    'rank': 1
    'activity_name': 'SSD2',
    'official_release_date': 2/1/2018,
    'weight': 71,
    'ram_support': 2GB DDR4 SDRAM,
    'storage': 4TB,
    'uuid': 1,
    'justification': 'Ram is the same but storage is futher apart
    than SSD1. Still more similar than SSD3 because SSD3 is using DDR3.'
}}]
```

```
### Response Format ###
[{{
    'rank',
    'official_release_date',
    'weight',
    'ram_support',
    'storage',
    'uuid',
    'justification'
}}]

Provide only the JSON list as the output, without any additional
text.
```

D.3 GENERATION PROMPT

```
### Instructions ###
Here is the multi-level BOM of one of the products you chose. Your
goal is to modify and adapt it to the TEST PRODUCT. Scale non-
electronic parts based on their attribute differences, but leave
electronic parts the same. Make sure the BOM you generated is
complete and contains all components that are typically in the
product. Create the new BOM in JSON list format. Add justification
for each row.

Attaibutes of the product you chose:

{product attributes in markdown}

Multi-level BOM of chosen attribute:
{BOM from BOM library}


Provide only the JSON list as the output, without any additional
text.
```

E   NAIVE LLM PROMPT

```
### Instructions ###
You are a researcher at environmental protection agency. you want
to do a life cycle assessment (LCA) of products based on their Bill
of Materials (BOM). You are interested in the environmental impact
associated with the materials and manufacturing phase of the
product. The product for analysis is a {product type} with the
following description: {product attributes}. Your goal is to
create a multi-level BOM for the given product with a hierarchical
structure that lists the components and subassemblies required to
make the product with correct quantities. Make sure you include
all necessary parts of the product.

Make the most of the given information. DO NOT say that
information is limited.

Avoid filler words such as "Based on the details" or "happy to
assist", keep your response to the point.
Do not repeat the given instructions or information.
```

```
Return your constructed BOM with all fields in the repository as
a json list.

### Output Schema ###
[{{
    'part_level',
    'part_type',
    'part_description',
    'qty',
    'material_or_component_type',
    'material_percent_recycled_content',
    'primary_manufacturing_process',
    'mass_g_per_unit',
    'method_for_weight',
    'primary_mfg_country',
    'ic_process_node_nm',
    'ic_die_size_mm2',
    'ic_package_type',
    'pcb_area_cm2',
    'pcb_type',
    'pcb_layers'
}}]

### Output Requirement ###
1. Start with part level 0 for the final assembly

Provide only the JSON list as the output, without any additional
text.
```

F METRIC DEFINITIONS

F.1 MAPE

MAPE - We use MAPE to measure the overall accuracy of IB BOM generation. MAPE is a statistical measure used to assess the accuracy of a model or forecasting method. It represents the average of absolute percentage errors between actual and predicted values. In our experiment context, the formula for calculating MAPE:

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \frac{|A_i - G_i|}{A_i} \times 100$$

Where:

A_i represents the calculated impact from the actual BOMs (TechInsights teardown),

G_i represents the calculated impact from the IB generated BOMs, and

n is the number of BOMs.

F.2 WEIGHTED $F1$

Material Weighted Precision and Recall - To evaluate the model's ability to generate a complete BOM structure with relevant materials and components, we use precision and recall. These metrics are adjusted based on the carbon impact associated with each material, giving higher weight to components with a larger environmental footprint. Unlike MAPE, precision and recall focus on whether the model correctly identifies high materiality BOM items, rather than accurately predicting their quantities. This ensures that critical, high-impact materials are prioritized in the evaluation.

$$Weighted\ Precision = \frac{\sum_{i=1}^{q} L_i \times G_i}{\sum_{i=1}^{q} G_i}$$

$$Weighted\ Recall = \frac{\sum_{i=1}^{p} L_i \times A_i}{\sum_{i=1}^{p} A_i}$$

$$Weighted\ F1 = 2 \times \frac{Weighted\ Precision \times Weighted\ Recall}{Weighted\ Precision + Weighted\ Recall}$$

Where:

L_i is a binary label representing whether material i is in both actual and generated BOM (1) or not (0),

G_i represents the calculated impact from the generated BOMs,

A_i represents the calculated impact from the actual BOMs (TechInsights teardown),

p is the number of materials/components in actual BOMs, and

q is the number of materials/components in generated BOMs.

The weighted $F1$ score is 99.5%. This means model generated BOMs include 95%+ carbon weighted components for almost all products.

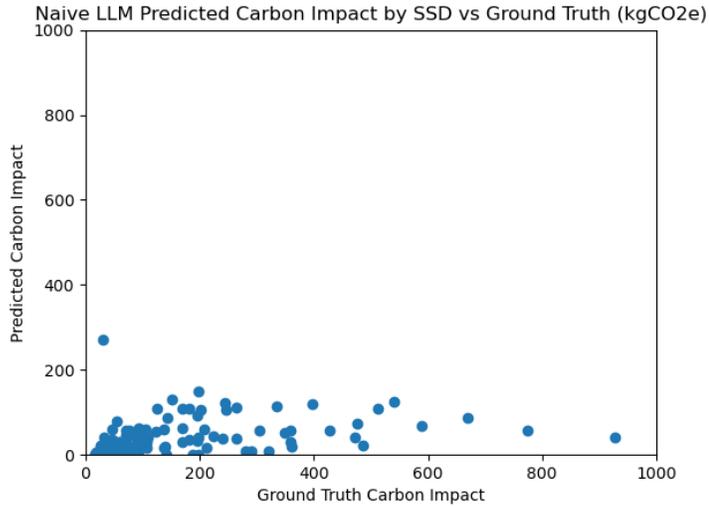### F.3 DISTRIBUTION OF PREDICTED CARBON EMISSION VS GROUND TRUTH

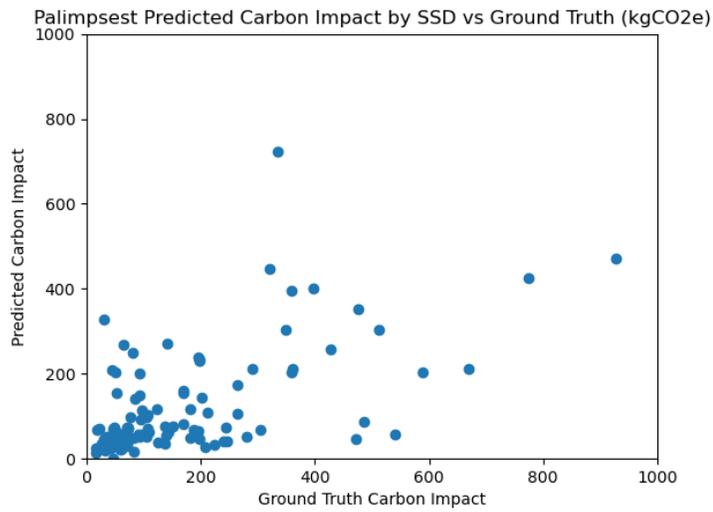Figure 3: Naive LLM Predicted Carbon Impact by SSD vs Ground Truth

Figure 4: Palimpsest Predicted Carbon Impact by SSD vs Ground Truth.