sustainability

MDPI

Article

PCF-RWKV: Large Language Model for Product Carbon Footprint Estimation

Zhen Li [1,†], Peihao Tang [1,†], Xuanlin Wang [1,†], Xueping Liu [1,*] and Peng Mou [2,*]

[1] Division of Advanced Manufacturing, Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China; zhen-li22@mails.tsinghua.edu.cn (Z.L.); tph22@mails.tsinghua.edu.cn (P.T.); wang-xl23@mails.tsinghua.edu.cn (X.W.)
[2] Department of Mechanical Engineering, Tsinghua University, Beijing 100084, China
* Correspondence: liuxp@sz.tsinghua.edu.cn (X.L.); moupeng@tsinghua.edu.cn (P.M.)
† These authors contributed equally to this work.

Abstract: As global climate change intensifies, assessing product carbon footprints serves as a foundational step for quantifying greenhouse gas emissions throughout a product's lifecycle, forming the basis for achieving sustainability and emission reduction goals. Traditional lifecycle assessment methods face challenges such as subjective boundary definitions and time-consuming inventory construction. This study introduces PCF-RWKV, a novel model based on the RWKV architecture with task-specialized low-rank adaptations (LoRAs). Trained on carbon footprint datasets, the model minimizes memory use and data interference, enabling efficient deployment on consumer-grade GPUs without relying on cloud computing. By integrating multi-agent technology, PCF-RWKV automates the creation of lifecycle inventories and aligns production processes with emission factors to calculate carbon footprints. This approach significantly improves the efficiency and security of corporate carbon footprint assessments, providing a potential alternative to traditional methods.

Keywords: product carbon footprint; lifecycle assessment; large language model; RWKV architecture; low-rank adaptation; multi-agents

check for updates

Academic Editor: Firoz Alam

Received: 18 December 2024
Revised: 20 January 2025
Accepted: 2 February 2025
Published: 6 February 2025

Citation: Li, Z.; Tang, P.; Wang, X.; Liu, X.; Mou, P. PCF-RWKV: Large Language Model for Product Carbon Footprint Estimation. *Sustainability* **2025**, *17*, 1321. https://doi.org/10.3390/su17031321

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As global climate change and its adverse impacts become increasingly severe, achieving sustainable development and reducing greenhouse gas emissions have become urgent global tasks. China has proposed the "3060" targets, aiming to peak carbon emissions by 2030 and achieve carbon neutrality by 2060, integrating these goals into the national strategy [1].

Accurate carbon accounting is essential for mitigation and emission-reduction strategies. The Intergovernmental Panel on Climate Change (IPCC) developed the Emission Factor Method (EFM) [2], and life cycle analysis (LCA) based on it has become a recognized carbon-accounting method with the help of ISO and IPS standards. Product development, a major source of greenhouse gas emissions, is now a focus. In 2023, China issued the Implementation Plan on Establishing a Carbon Footprint Management System, which specifies the tasks of building a carbon footprint management system for Chinese products in the coming period. Product carbon footprint (PCF) is a crucial assessment tool [3,4]. LCA, a traditional PCF-evaluation method, has challenges. Gathering lifecycle inventory data is labor-intensive, costly, and inefficient due to the complexity and variety of data at each product stage [5–7]. Also, defining product boundaries based on expert knowledge

adds uncertainty and subjectivity, affecting assessment results and decision-making. Thus, efficient PCF assessment under unified standards is crucial.

Large language models (LLMs) have emerged, addressing the inefficiencies and high costs of traditional LCA methods and enabling rapid content generation based on expert knowledge [8]. However, current Transformer-based general-purpose models face issues [9]. Cloud-deployed models risk data leakage, while local deployment is hampered by hardware limitations due to high memory and computational demands. These models also lack domain-specific knowledge for carbon footprint accounting, leading to inaccurate responses.

Our research introduces the innovative PCF-RWKV model, focusing on the RWKV architecture, low-rank adaptation (LoRA) training, and multi-agent technology integration. This overcomes the limitations of existing methods like AutoPCF [10] and traditional Transformer models. AutoPCF lacks targeted adjustments, task-specific training, and agent technology. In contrast, PCF-RWKV uses low-rank adaptation, is trained on a 7-billion-parameter specialized dataset for carbon footprint assessment, and can be efficiently deployed locally on a single consumer-grade GPU, ensuring data security.

Through multi-agent technology, PCF-RWKV can automatically generate models for product production, raw material energy use, and waste emissions, effectively constructing a "cradle-to-gate" lifecycle model.

The "Cradle-to-gate" lifecycle model [11] evaluates a product's environmental impact starting from raw material acquisition through to the point it exits the manufacturing facility. It quantifies carbon footprint by assessing energy use, greenhouse gas emissions, and additional environmental factors, excluding transportation and product usage phases. It utilizes semantic-based matching to assign appropriate emission factors for each production process, allowing for accurate carbon footprint calculations. This offers enterprises a secure, efficient, and professional solution for product carbon footprint (PCF) assessment.

The main contributions of this paper are as follows:

1. The PCF-RWKV carbon footprint assessment model was designed based on the RWKV framework. The model incorporates multiple stacked residual blocks within the RWKV architecture and three task-specialized LoRAs, enabling efficient operation on low-power edge devices while ensuring data processing security and model performance stability. Compared to traditional Transformer models, the PCF-RWKV based on the RWKV architecture reduces time complexity from $O(n^2)$ to $O(n)$ and space complexity from $O(n^2 + n)$ to $O(1)$, achieving constant memory usage and significantly enhancing the ability to process long sequence data.
2. Low-rank adaptive techniques were applied to train the model on a product carbon footprint assessment knowledge dataset. Through refined parameter allocation and multi-head LoRA task-specialized training, the model reduces data interference during training and significantly decreases memory waste caused by redundant base model deployments.
3. A multi-agent collaborative framework for carbon footprint assessment was developed based on the AutoGen framework. This framework facilitates dynamic knowledge base updates and automated carbon footprint assessment processing through agent collaboration, significantly enhancing the model's practicality and adaptability.
4. The effectiveness of the proposed method was experimentally verified. The experimental results indicate that, compared to traditional methods, PCF-RWKV offers significant advantages in assessment efficiency and resource consumption. Current shortcomings in the accuracy and stability of PCF-RWKV were also analyzed, and directions for future improvements were suggested.

The structure of this paper is as follows: Section 1 introduces the background of product carbon footprint assessment and the main contributions of this study. Section 2 reviews the current major methods and related research in the field of PCF assessment. Section 3 details the architectural design and training methods of the PCF-RWKV model and explains the design and collaboration process of the multi-agent collaborative framework. Section 4 evaluates the model's performance through experiments. Section 5 provides an in-depth discussion of the experimental results and points out future research directions. Finally, Section 6 concludes the paper.

2. Background and Related Work

The calculation of product carbon footprint (PCF) typically follows the lifecycle assessment (LCA) process, which generally includes five stages: raw material extraction, manufacturing, distribution, use, and end-of-life disposal. This comprehensive calculation is often referred to as "cradle-to-grave". In this study, the PCF calculation scope is limited to "cradle-to-gate", primarily focusing on the production and manufacturing process, as these are the lifecycle stages with the highest concentration of raw material and energy consumption.

Estimating a product's carbon footprint usually involves three main steps: defining the system boundary, conducting a life cycle inventory analysis, and assessing environmental impacts.

Lifecycle assessment (LCA) experts determine the system boundary based on the definition of the product's carbon footprint and build a life cycle inventory (LCI) using their professional knowledge and experience. Specifically, experts need to analyze each stage of the product lifecycle and identify the inputs and outputs required to construct the product lifecycle model. Subsequently, they assess the environmental impacts of all inputs and outputs throughout the product lifecycle, focusing on greenhouse gas emissions. Inputs typically include raw materials and energy, as well as emissions from the production processes of raw materials, consumables, and auxiliary energy (such as electricity). Outputs generally include waste gases, wastewater, and waste materials, such as greenhouse gas emissions from fossil fuel use. Finally, these emissions are converted into carbon dioxide equivalents (CO_2-eq) and summed to calculate the product's carbon footprint.

This process can be time-consuming, involving the decomposition of each lifecycle stage of the product based on industry production experience or relevant literature to determine the inputs and outputs for each process. The input–output inventory values are then determined based on actual production conditions, and the results are calculated. Despite the existence of some lifecycle modeling tools and databases that can help improve calculation efficiency, completing a product carbon footprint assessment still requires several days or even months.

Thus, developing efficient and accurate carbon footprint assessment tools is crucial. Improving assessment efficiency via technology has become a hot topic among scholars.

González-González et al. (2022) presented an explainable machine learning approach [12] to automatically estimate industrial carbon footprints. By classifying bank transaction records and using NLP on descriptions, it overcomes the inefficiencies of existing protocols. The method achieved good metrics and provided interpretable decision-making ways. Zeng et al. (2023) proposed an online accounting method [13] for coal mine carbon emissions in coal industry groups. Using a B/S architecture, it changed from top-down to bottom-up. This allows online data handling and publication, enhancing accounting efficiency and accuracy, and enabling result sharing.

Deng et al. (2023) presented the AutoPCF framework [10] using large language models (LLMs) to automate product carbon footprint (PCF) accounting. By testing five

LLMs' ability to simulate the product lifecycle, we found that it could significantly reduce modeling time from days to minutes for automatic PCF modeling and estimation.

Wang and Luo (2023) proposed a carbon emission accounting and monitoring system [14] with AI algorithms, optimizing accounting via algorithm analysis, applying three combined models (particle swarm optimization with 90% stability and 89% accuracy) for multi-level real-time monitoring and prediction to aid carbon-reduction strategies. Huang and Mao (2024) put forward a data-driven approach [15] using AI to manage global supply chain carbon footprints, constructing an AI-centered framework to analyze, optimize, and monitor footprints with machine learning and optimization on relevant datasets for reduction strategies.

Building on these studies, this paper designs and trains the novel dual-carbon large model PCF-RWKV based on the RWKV architecture. The PCF-RWKV model, trained with low-rank adaptive techniques on a dual-carbon dataset, comprises 7 billion parameters and can be efficiently deployed on a consumer-grade single GPU locally, ensuring enterprise data security. By integrating multi-agent technology, PCF-RWKV automatically generates models for product production processes, raw material energy use, and waste emissions, constructs lifecycle models, and ultimately calculates the carbon footprint, enhancing the accuracy and stability of product carbon footprint evaluations using large language models.

This paper's work represents a significant advancement in the field by combining the strengths of previous studies. Like González-González et al., we prioritize explainability and trust in our model's outputs. We adopt the efficiency and speed improvements showcased by Deng et al. and Wang and Luo, and we echo Huang and Mao's comprehensive approach to managing carbon footprints across supply chains. Our PCF-RWKV model not only streamlines the process of carbon footprint assessment but also ensures data security and enhances the scalability of carbon footprint evaluations, making it a robust tool for enterprises in their pursuit of sustainable development and carbon reduction strategies.

3. Materials and Methods

3.1. Architecture of the PCF-RWKV Model

To prevent the model from crashing due to excessive memory usage during the deployment of large Transformer models on edge devices, the PCF-RWKV model adopts the RWKV architecture as its foundational structure. RWKV [16] (Receptance-Weighted Key Value) is a novel architecture that combines the parallel computing capabilities of Transformers [17] with the time series processing capabilities of RNNs [18], featuring linear computational complexity and supporting parallelized training. As illustrated in Figure 1, the PCF-RWKV model consists of multiple stacked residual blocks within the RWKV architecture and three task-specialized LoRA (low-rank adaptation) heads. Each residual block comprises a time-mixing sub-block and a channel-mixing sub-block, which are embedded with recurrent structures to leverage historical information. This model architecture combines the advantages of RNNs and Transformers while avoiding their respective major drawbacks. The three task-specialized LoRA heads [19,20] are designed to handle the LCI generation task, multi-agent task orchestration, and product carbon footprint calculation task, respectively. This specialized approach reduces data interference during training and significantly minimizes memory waste caused by redundant base model deployments during model deployment.

The model employs a unique attention-like mechanism, incorporating a time-dependent softmax operation to enhance numerical stability and mitigate the vanishing gradient problem. This ensures that gradients propagate along the most relevant paths. Additionally, the built-in layer normalization within the model structure aids in stabilizing gradients, effectively addressing both vanishing and exploding gradient issues. These design elements not only

accelerate the training speed of deep neural networks but also facilitate multi-layer stacking, allowing the model to capture complex relationships at different levels of abstraction, thereby surpassing the performance of traditional RNN models.

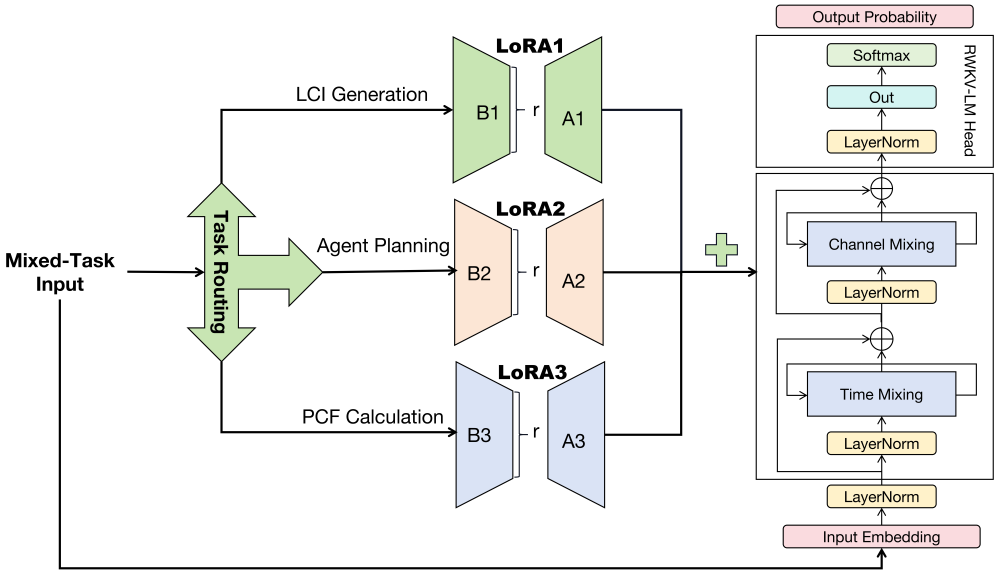


Figure 1. PCF-RWKV architecture diagram.

The RWKV model introduces a linear attention mechanism, replacing the dot-product-based attention computation in traditional Transformer models. This design not only improves data processing efficiency but also enhances the model's ability to capture key information by focusing on the channel features of the data, maintaining low computational complexity without sacrificing model performance. Compared to traditional Transformer models, the PCF-RWKV based on the RWKV architecture reduces time complexity from $O(n^2)$ to $O(n)$ and space complexity from $O(n^2 + n)$ to $O(1)$ [16].

In handling sequential data, RWKV incorporates a time-mixing block to achieve global interaction, akin to the self-attention mechanism in Transformer models. The core parameters of the time-mixing block include R, W, K, and V, which are used in a novel weighting and mixing approach to capture long-range dependencies in sequential data. The functions of these four core parameters are as follows:

- R (Receptance): This parameter facilitates memory of past information and implements a forgetting mechanism through the Sigmoid activation function. R helps the model decide which past information to retain or forget at the current timestep.
- W (Weight): This parameter assigns weights to different positions in the sequence based on their relative positions. W is a trainable parameter within the model.
- K (Key) and V (Value): In the RWKV model, K and V are analogous to the K and V in the attention mechanism, representing the key and value information at different positions in the sequence.

These four core elements interact multiplicatively at each timestep, as illustrated in Figure 2.

Unlike traditional Transformer models that directly use the current token's embedding as input, the RWKV model employs a weighted combination of the current token and the previous token's embeddings as input. This computation method dynamically integrates information from consecutive tokens in the sequence by assigning different weights to token embeddings at different timesteps, providing the model with richer contextual information and enhancing its ability to capture sequence dependencies. In RWKV, while the computation methods for R, K, and V differ from the traditional attention mechanism,

their basic functions remain similar—capturing the relationships between tokens at different positions in the sequence. In this architecture, all vectors involved in linear projections (R, K, V in time mixing and R′, K′ in channel mixing) are generated through linear interpolation between the inputs at the current and previous timesteps. The computation is as follows:

$$r_t = W_r \cdot (\mu_r \odot x_t + (1 - \mu_r) \odot x_{t-1}) \tag{1}$$

$$k_t = W_k \cdot (\mu_k \odot x_t + (1 - \mu_k) \odot x_{t-1}) \tag{2}$$

$$v_t = W_v \cdot (\mu_v \odot x_t + (1 - \mu_v) \odot x_{t-1}) \tag{3}$$

The implementation of the attention mechanism in the RWKV model diverges significantly from traditional Transformer methods. While traditional Transformers rely on absolute position biases, RWKV abandons this approach and adopts a relative position encoding mechanism. In traditional models, during training, an independent bias parameter $w_{i,j}$ has to be learned for each position in the sequence. In contrast, RWKV uses a shared parameter vector for position encoding. This change not only reduces the model complexity remarkably but also boosts the parameter learning efficiency.

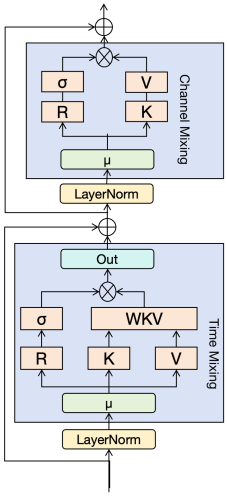


Figure 2. Time-mixing module and channel-mixing module.

The calculation of the WKV operator in RWKV is somewhat similar to that in the Attention-Free Transformer (AFT) [21]. However, there is a key difference. In AFT, W is a pairwise matrix, while in the RWKV model, W is a channel-level vector that is modified by relative positions. This recursive behavior in the RWKV model is defined by the time-dependent update of the WKV vector. To avoid potential degradation of W, a U vector is introduced, which focuses solely on the current token.

The time-dependent update of the WKV vector is formalized by the following formula:

$$wkv_t = \frac{\sum_{i=1}^{t-1} e^{-(t-1-i)w+k_i} \odot v_i + e^{u+k_t} \odot v_t}{\sum_{i=1}^{t-1} e^{-(t-1-i)w+k_i} + e^{u+k_t}} \tag{4}$$

The numerator represents the weighted sum of past and current key-value pairs. The term $\sum_{i=1}^{t-1} e^{-(t-1-i)w+k_i} \odot v_i$ accumulates the contributions of previous tokens. Here, $e^{-(t-1-i)w+k_i}$ serves as a weight coefficient, determining the contribution degree of the i-th previous token's key-value pair (k_i, v_i) to the current wkv value. The exponential part $-(t-1-i)w$ encodes the relative position of the (i)-th previous token with respect to the current token t, where w is the factor related to relative positions in the W vector. Intuitively, for previous tokens farther from the current token, the value of the exponential part is more

negative, meaning their impact on the current wkv value is relatively small. Conversely, previous tokens closer to the current token have a greater impact.

The term $e^{u+k_t} \odot v_t$ calculates the contribution of the current token. The U vector, through u in the exponent, provides a separate enhancement for the key-value pair (k_t, v_t) of the current token. This ensures that the model can focus more on the current information and not be overwhelmed by historical information, thus striking a balance between historical and current information.

The denominator $\sum_{i=1}^{t-1} e^{-(t-1-i)w+k_i} + e^{u+k_t}$ normalizes the weighted sum in the numerator. This normalization operation guarantees that the resulting wkv_t value is a reasonable weighted average of past and current key-value pairs. It is crucial for the stable operation of the model, enabling the model to output relatively stable and comparable results when processing sequence data of different lengths.

The attention mechanism in the RWKV model can be expressed in a recursive form, as shown in Figure 3. This recursive form endows the model with several advantages. It allows the model to inherit the linear time complexity $O(n)$ of linear Transformers, ensuring high-speed computation when dealing with long-sequence data. Meanwhile, it retains the simplicity of AFT, making the model's structure and computational process relatively straightforward. Through this recursive expression, the model improves its computational efficiency while maintaining a strong expressive power in handling complex sequence data. By this means, the RWKV model enhances its sensitivity and ability to capture key information in sequence data while maintaining computational efficiency.

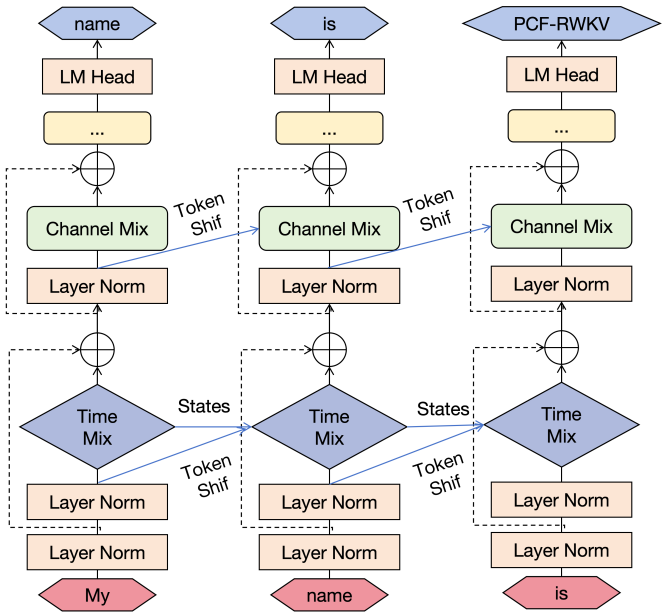


Figure 3. Token shift processing of RWKV.

The RWKV architecture represents a significant advancement in neural network design by effectively integrating the strengths of Transformers and RNNs, thereby overcoming the inherent limitations of each.

Unlike traditional RNNs, the RWKV architecture not only preserves the performance advantages of Transformer models but also enables more stable gradient propagation. When compared with traditional Transformer models, the RWKV architecture offers remarkable improvements in computational efficiency. It reduces the time complexity from $O(n^2)$ to $O(n)$ and the space complexity from $O(n^2 + n)$ to $O(1)$, achieving constant memory usage [16]. These optimizations endow the PCF-RWKV model with enhanced performance when handling long-sequence problems, such as data analysis.

During the inference phase, both the computation time and memory requirements of RWKV grow linearly. This is in sharp contrast to the quadratic growth exhibited by traditional Transformer models. The following Table 1 provides a clear comparison of the inference complexity between RWKV and other models:

The linear growth in inference complexity of RWKV makes it a more efficient choice, especially for applications that involve processing long-sequence data. This efficiency, combined with its ability to inherit the beneficial aspects of both RNNs and Transformers, positions the RWKV architecture as a promising solution in the field of neural network research and application.

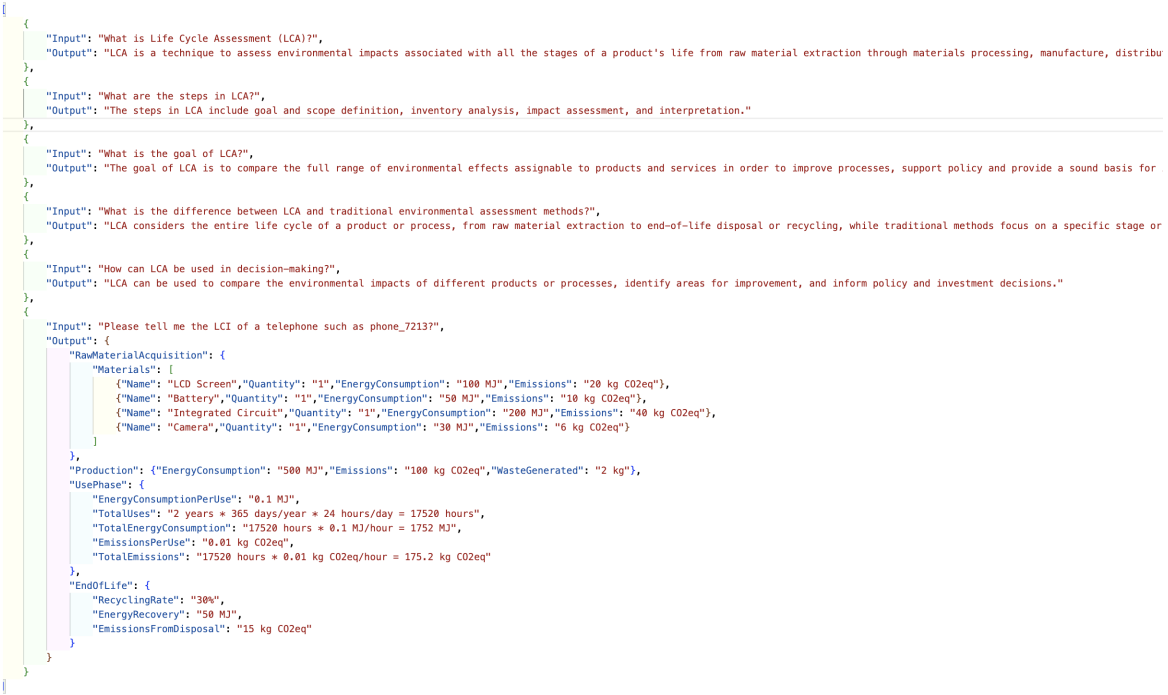
Table 1. Comparison of inference complexity and memory requirements.

Model	Inference Complexity	Memory Requirement
Transformer	$O(T^2 d)$	$O(T^2 + Td)$
Reformer	$O(T \log T d)$	$O(T \log T + Td)$
Performer	$O(Td^2 \log d)$	$O(Td \log d + d^2 \log d)$
RWKV (This Model)	$O(Td)$	$O(d)$

3.2. Construction and Data Augmentation of the PCF-RWKV Training Dataset

Despite large language models achieving excellent results in various benchmarks, they lack the in-depth knowledge and data support specific to carbon footprint assessment, preventing their direct application in this field. Therefore, customizing model training with domain-specific data is crucial to enhance performance and accuracy in carbon footprint evaluation.

The dataset for model training is formatted in JSON, comprising question inputs and answer outputs, as shown in Figure 4.



```
[
  {
    "Input": "What is Life Cycle Assessment (LCA)?",
    "Output": "LCA is a technique to assess environmental impacts associated with all the stages of a product's life from raw material extraction through materials processing, manufacture, distribu
  },
  {
    "Input": "What are the steps in LCA?",
    "Output": "The steps in LCA include goal and scope definition, inventory analysis, impact assessment, and interpretation."
  },
  {
    "Input": "What is the goal of LCA?",
    "Output": "The goal of LCA is to compare the full range of environmental effects assignable to products and services in order to improve processes, support policy and provide a sound basis for
  },
  {
    "Input": "What is the difference between LCA and traditional environmental assessment methods?",
    "Output": "LCA considers the entire life cycle of a product or process, from raw material extraction to end-of-life disposal or recycling, while traditional methods focus on a specific stage or
  },
  {
    "Input": "How can LCA be used in decision-making?",
    "Output": "LCA can be used to compare the environmental impacts of different products or processes, identify areas for improvement, and inform policy and investment decisions."
  },
  {
    "Input": "Please tell me the LCI of a telephone such as phone_7213?",
    "Output": {
      "RawMaterialAcquisition": {
        "Materials": [
          {"Name": "LCD Screen","Quantity": "1","EnergyConsumption": "100 MJ","Emissions": "20 kg CO2eq"},
          {"Name": "Battery","Quantity": "1","EnergyConsumption": "50 MJ","Emissions": "10 kg CO2eq"},
          {"Name": "Integrated Circuit","Quantity": "1","EnergyConsumption": "200 MJ","Emissions": "40 kg CO2eq"},
          {"Name": "Camera","Quantity": "1","EnergyConsumption": "30 MJ","Emissions": "6 kg CO2eq"}
        ]
      },
      "Production": {"EnergyConsumption": "500 MJ","Emissions": "100 kg CO2eq","WasteGenerated": "2 kg"},
      "UsePhase": {
        "EnergyConsumptionPerUse": "0.1 MJ",
        "TotalUses": "2 years * 365 days/year * 24 hours/day = 17520 hours",
        "TotalEnergyConsumption": "17520 hours * 0.1 MJ/hour = 1752 MJ",
        "EmissionsPerUse": "0.01 kg CO2eq",
        "TotalEmissions": "17520 hours * 0.01 kg CO2eq/hour = 175.2 kg CO2eq"
      },
      "EndOfLife": {
        "RecyclingRate": "30%",
        "EnergyRecovery": "50 MJ",
        "EmissionsFromDisposal": "15 kg CO2eq"
      }
    }
  }
]
```


Figure 4. PCF-RWKV training dataset example.

Currently, high-quality PCF datasets for carbon footprint evaluation are relatively scarce. Synthetic data [22,23] is a common method for dataset construction when data is insufficient. Large language models can generate consistent training data results based

on their text comprehension capabilities across various file types. To address this issue, a strategy combining large language model capabilities with human-in-the-loop feedback supervision was employed for data synthesis and augmentation.

The process of training data synthesis and enhancement through large models and human feedback is clearly illustrated in Figure 5. Initially, the publicly accessible academic literature, industry standards, production data, and carbon footprint evaluation cases in the dual-carbon field were extensively collected to form the initial seed datasets. These seed datasets were input into large language models, and through carefully crafted prompts, JSON datasets meeting the required training format and content were generated. Throughout this process, manual sampling inspections were conducted to ensure the quality and compliance of the generated data. Based on the inspection results, seed data and prompts were iteratively optimized to improve the quality and coverage of data synthesis.

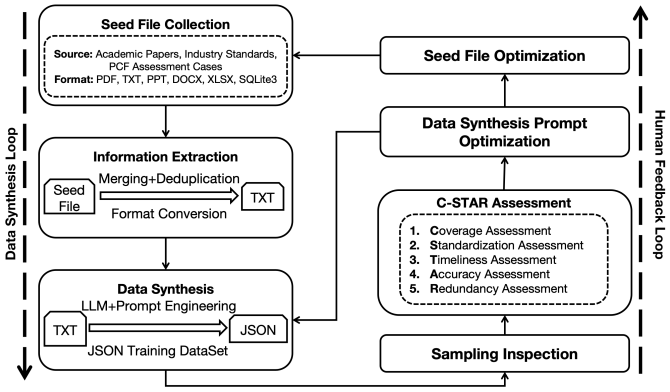


Figure 5. Data synthesis and enhancement strategy based on large models and human feedback.

In this process, the C-STAR assessment mechanism was implemented to focus on multi-dimensional data quality evaluation. C-STAR is a comprehensive evaluation framework designed to ensure the integrity and utility of the synthesized data. It includes the following dimensions:

1. **C**overage Assessment: Ensuring that the dataset encompasses all necessary knowledge areas and application scenarios.
2. **S**tandardization Assessment: Verifying the consistency and uniformity of data format and content.
3. **T**imeliness Assessment: Checking the update frequency and real-time nature of the data to ensure its current relevance.
4. **A**ccuracy Assessment: Ensuring the correctness and reliability of the data, minimizing errors and misleading information.
5. **R**edundancy Assessment: Identifying and eliminating duplicate data to enhance the efficiency and compactness of the dataset.

Through a human feedback loop, data synthesis prompts and seed files are continuously optimized, improving the dataset's quality. This strategy successfully constructed a PCF dataset rich in model Q&A training data, providing abundant resources for model training.

3.3. Low-Rank Adaptation Training of the PCF-RWKV Model

Various techniques exist for training large language models. This study employs the low-rank adaptation (LoRA) algorithm to train the large model in the PCF domain. LoRA [24] (low-rank adaptation) is an efficient parameter tuning technique that reduces the number of parameters to learn by constraining the update matrices to be low-rank.

This method leverages matrix factorization to decrease the parameter learning volume, employs specific parameter initialization strategies and fine-tuning processes, freezes the original weights, and trains only the low-rank matrices. The core of the LoRA algorithm lies in optimizing the pre-trained model weights. By freezing the weights of the pre-trained model and introducing trainable low-rank matrices to adjust model parameters, LoRA significantly reduces the number of trainable parameters required for downstream tasks. Specifically, this method decomposes the original weight matrix into the product of two lower-rank matrices, where one matrix's rank is much lower than the other, using only the low-rank matrices for computation. This strategy not only reduces the number of model parameters that need to be trained, improving training efficiency, but also maintains model performance while avoiding increased inference latency.

As shown in Figure 6, The rank decomposition matrix is a critical component in the LoRA framework, consisting of two linear projections designed to reduce and subsequently restore the dimensionality of the input. This mechanism involves adding the output from these projections to the output derived from the model's pre-trained weights. The resulting updated layer, formed by integrating these parallel transformations, effectively captures and learns the necessary updates to the underlying layer's weights. This approach enables efficient adaptation of the model while maintaining computational efficiency.

$$W_{\text{ft}} = W_{\text{pt}} + \Delta W = W_{\text{pt}} + AB \tag{5}$$

$$W_{\text{ft}}, W_{\text{pt}}, \Delta W, AB \in \mathbb{R}^{d \times d} \tag{6}$$

$$A \in \mathbb{R}^{d \times r}, \quad B \in \mathbb{R}^{r \times d} \tag{7}$$

The matrix product AB matches the dimensionality of a full fine-tuning update, yet by decomposing the update into two smaller matrices, LoRA ensures a low-rank structure that significantly reduces the number of parameters requiring optimization. Rather than fine-tuning all parameters in the pre-trained model layers, LoRA focuses exclusively on optimizing the rank decomposition matrix. This approach approximates the effects of full fine-tuning. Matrix A is initialized with small random values, while B is initialized to zero, preserving the integrity of the model's original pre-trained weights at the onset of fine-tuning.

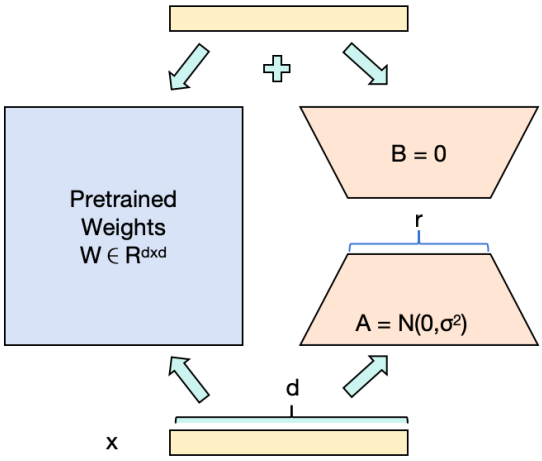


Figure 6. Low-rank adaptation schematic diagram.

Empirical evidence indicates that setting the LoRA rank r to match the rank of the pre-trained weight matrices can effectively approximate the expressiveness of full fine-tuning. Increasing the rank r enhances the accuracy of this approximation, allowing

the low-rank adaptation to more closely mimic the updates achieved through full fine-tuning. However, in practice, remarkably low values of r are often sufficient. This enables substantial reductions in computational and memory demands while resulting in minimal performance degradation. For instance, LoRA can fine-tune models such as GPT-3 using merely 0.01% of the total parameters, all while maintaining performance levels comparable to those achieved through full fine-tuning.

The core of the LoRA method is represented by the following equation:

$$W_{\text{ft}} = W_{\text{pt}} + \frac{\alpha}{r} AB \tag{8}$$

In Equation $W_{\text{ft}} = W_{\text{pt}} + \frac{\alpha}{r} AB$, W_{ft} are the fine-tuned weights, W_{pt} are the pre-trained weights, and AB is the low-rank update. A and B are low-rank matrices for task-specific weight adjustments, and r normalizes the low-rank update's impact.

The scaling factor α is key for integrating the low-rank update with pre-trained weights. By default, $\alpha = 1$, balancing the influence of pre-trained and updated weights. This balance helps the model keep generalization while adapting to new tasks.

Furthermore, α can be adjusted. Empirical studies show larger α benefits models with higher r. This adjustment improves LoRA's adaptation ability, letting the model use complex data relationships without full traditional fine-tuning costs.

The training environment was deployed on a cloud server equipped with an NVIDIA A100 GPU with 40GB of VRAM (NVIDIA, Santa Clara, CA, USA). Training parameters utilized the BFloat16 (bf16) data type, balancing computational precision with model performance and memory usage efficiency. The context length was set to 512 to ensure the model could adequately capture sequence information. Given the dataset size, 100 iterative steps per epoch were scheduled, starting from epoch 0 and planned up to epoch 500, with model states saved every 5 epochs for subsequent evaluation and analysis. The Adam optimizer was used, configured with hyperparameters: Epsilon set to 1×10^{-8}, Beta 1 at 0.9, and Beta 2 at 0.999, ensuring stability and efficiency in the optimization process. To alleviate initial training instability, a learning rate warm-up strategy was implemented, gradually increasing the learning rate over the first 1000 iterations. The initial learning rate was set at 0.001, gradually decreasing to 1×10^{-5} for finer parameter adjustments.

Based on the meticulously synthesized dual-carbon dataset, the RWKV architecture's dual-carbon base model was successfully trained, featuring 7 billion parameters capable of smooth operation on consumer-grade single-GPU setups, demonstrating good overall performance on the test set, with detailed test results to be presented in Section 4 of the paper.

3.4. PCF-RWKV Multi-Agent Architecture Design

While the PCF-RWKV base model provides an analytical understanding of carbon accounting-related information, integrating the PCF-RWKV large model into the product carbon footprint evaluation workflow necessitates combining it with multi-agent technology [25].

The PCF-RWKV employs the Autogen framework [26] to construct a collaborative agent network, breaking down various aspects of product carbon footprint accounting and assigning tasks to different agents. Through cooperation among agents, efficient product carbon footprint assessment support is achieved, as depicted in Figure 7.

The system comprises the following key agents:

1. **LCI Generation Agent**: The LCI Generation Agent is responsible for creating the life cycle inventory (LCI). This agent utilizes specialized knowledge extracted from the PCF training dataset to transform abstract knowledge into a life cycle inventory

specific to the user's input product. It analyzes the material and energy flows across all stages of the product's lifecycle, constructing a detailed and accurate LCI to provide a data model for carbon footprint quantification.

2. **Activity Data Generation Agent**: The Activity Data Generation Agent bases its operations on the LCI to generate activity data related to the product lifecycle. These data include information on material and energy consumption, as well as waste generated during production. By analyzing the activities of the product or semi-product at different stages, the agent produces detailed data for each activity, laying the groundwork for subsequent carbon footprint calculations.
3. **Emission Factor RAG Agent**: The emission factor RAG agent employs search-enhanced generation techniques to dynamically retrieve and generate applicable emission factors, drawing from an LCA document corpus and the latest research. It does not rely on a fixed database but instead dynamically generates optimized emission factors to ensure the accuracy of carbon footprint calculations.
4. **PCF Calculation Agent**: The PCF calculation agent combines activity data and emission factors to calculate the product's carbon footprint. Using the formula "Product Carbon Footprint = Emission Factor × Activity Data", this agent performs calculations and utilizes large language models (LLMs) to generate carbon footprint calculation programs tailored to the task's characteristics. It precisely quantifies carbon emissions at each stage of the product lifecycle, delivering reliable carbon footprint calculation results.

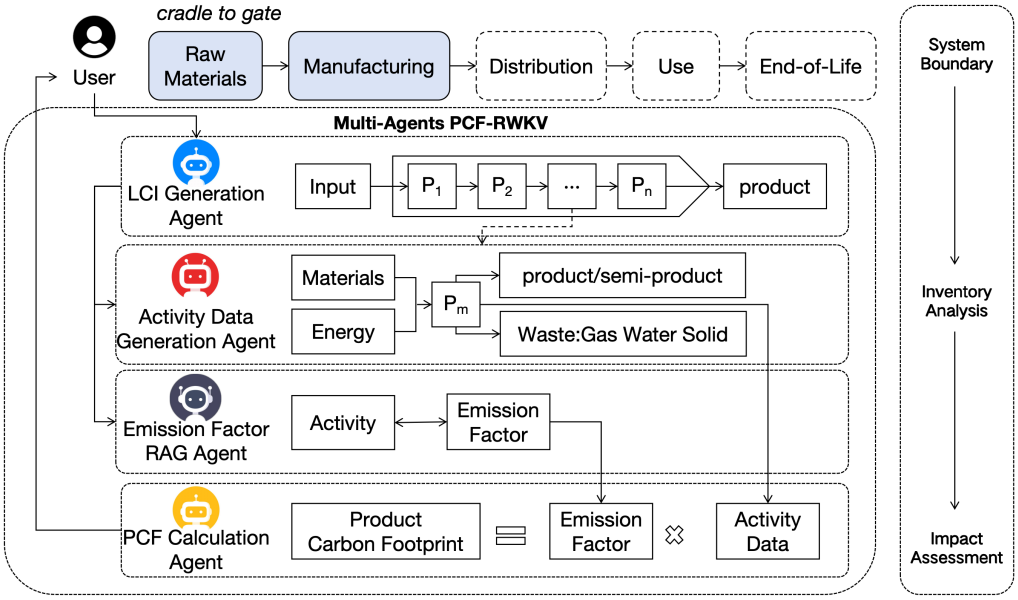


Figure 7. PCF-RWKV multi-agent architecture diagram.

In the LCA product carbon footprint analysis system, the agents collaborate meticulously to achieve efficient and precise carbon footprint evaluations. Initially, the task orchestration agent receives user input, utilizing natural language processing techniques to refine task requirements. It then instructs the LCA knowledge base agent to retrieve the relevant literature and research findings, providing theoretical support for subsequent steps. The LCI generation agent, using user information and expert knowledge, analyzes the material and energy flows throughout the product lifecycle to construct a comprehensive life cycle inventory (LCI). Based on this, the activity data generation agent further analyzes each stage's activities, producing detailed data on material and energy consumption. The emission factor RAG agent, employing search-enhanced generation techniques, dynamically retrieves and generates applicable emission factors to ensure data accuracy. Finally,

the PCF calculation agent combines activity data and emission factors, utilizing large language models to generate carbon footprint calculation programs. It precisely quantifies carbon emissions and feeds the results back to the task orchestration agent. This entire process, driven by the automated collaboration of agents, enhances evaluation efficiency and accuracy, providing a scientific basis for sustainable production and consumption.

4. Results

4.1. Model Evaluation for Product Carbon Footprint Assessment

Evaluating a model requires a comprehensive consideration of multiple aspects of its capabilities. The core objective is to test and compare the model's accuracy and execution efficiency within the context of product carbon footprint assessment. Accuracy and execution efficiency each contributes 50 points, for a total score of 100 points, which is used to measure the model's suitability for the product carbon footprint evaluation scenario.

Regarding model accuracy, the evaluation is conducted through test questions by comparing the model's scores. The evaluation questions cover the entire lifecycle of a product, from manufacturing to final disposal and recycling, ensuring the comprehensiveness of the assessment results. Due to the scarcity of carbon footprint-related test data, a large model-based generation method similar to the training dataset was employed to construct the test dataset. This method generated and screened 100 test questions centered around product carbon footprint evaluation as a benchmark. The questions are all true/false, comprising 50 correct and 50 incorrect statements. The original data sources for this test set include product carbon footprint evaluation cases from different industries and production processes, as well as common knowledge related to carbon footprints. These sources provide a basis for assessing the model's capabilities in the context of product carbon footprint evaluation.

For this evaluation, three metrics were selected to measure the model's prediction accuracy: precision, recall, and F1 score. Ultimately, the model's score in the accuracy test phase is determined by multiplying the F1 score by 50. These metrics are calculated based on the confusion matrix, which records the relationship between the model's predicted results and the actual results. In the confusion matrix[27], true positives (TPs) refer to the number of samples correctly predicted as positive by the model; true negatives (TNs) refer to the number of samples correctly predicted as negative; false positives (FPs) refer to the number of samples incorrectly predicted as positive; and false negatives (FNs) refer to the number of samples incorrectly predicted as negative. The specific calculation methods are as follows:

Recall: This metric is calculated by dividing the number of true positives by the sum of true positives and false negatives (cases that actually occurred but were not predicted by the model). The formula is:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

F1 Score: The F1 score is the harmonic mean of precision and recall, balancing the two to provide a comprehensive performance metric. It offers a more balanced measure of performance, particularly when there is a significant discrepancy between precision and recall, thus avoiding the limitations of a single metric. The formula is:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

where precision is calculated as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

In evaluating model execution efficiency, we focus on two key metrics: processing time and resource consumption. Processing time is the average time for the model to handle a sample or batch, reflecting its response speed. Resource consumption refers to the model's demand for CPU, GPU, and memory during operation. Lower resource-consuming models are more adaptable across different computing devices, enhancing applicability. We use fuzzy mathematics to score processing time and resource consumption. Resource consumption is classified into five levels (very low, low, medium, high, and very high) with scores of 5, 10, 15, 20, and 25, respectively. Similarly, processing time is divided into very short, short, medium, long, and very long, also with scores from 5 to 25.

We compared the performance of several models: baseline RWKV-7B, baseline PCF-RWKV-7B, open-source Qwen1.5-7b-chat, RWKV with agents, PCF-RWKV with agents, GPT-4, and GLM6B [28]. The results are in Table 2. During the assessment, we recorded scores of various metrics on the test set using automated scripts and manual blind reviews. Scripts logged processing time and resource consumption during runtime, and multiple runs were done to get average values for fair and objective scoring.

In the evaluation of model performance for product carbon footprint assessment, we conducted a systematic assessment of multiple models, analyzing their performance in Table 2, Figure 8 across various dimensions, including efficiency, accuracy, and overall score.

Firstly, in terms of efficiency score, the PCF-RWKV (7B-Agent) achieved a score of 35.00, indicating moderate resource utilization efficiency, while the GPT-4 (base) scored only 15.00, suggesting relatively poor utilization of computational resources. In contrast, the RWKV-LoRA-LCI model scored 40.00, demonstrating good performance and high efficiency in resource utilization. Both RWKV-LoRA-PCF and RWKV-LoRA-Plan also achieved an efficiency score of 40.00, reflecting similar levels of efficiency. Overall, the RWKV-LoRA series models generally outperformed GPT-4 (base) in terms of efficiency scores, indicating their advantages in resource utilization and computational efficiency.

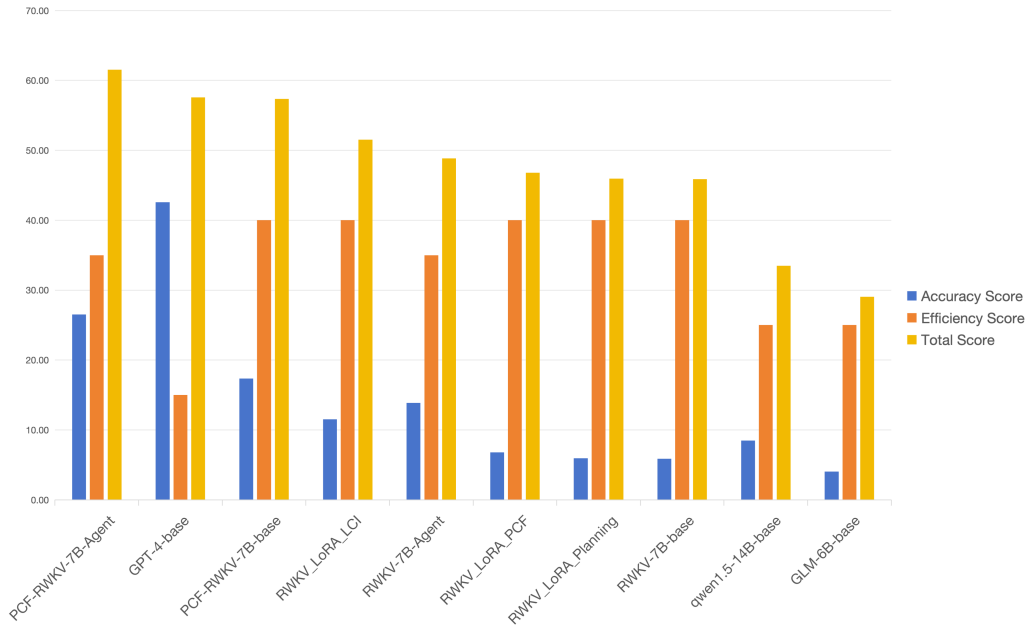


Figure 8. Model evaluation result for product carbon footprint assessment. (1) GPT-4 leads in accuracy score. (2) PCF-RWKV 7B base has the highest efficiency score. (3) PCF-RWKV 7B agent ranks top in overall total score.

Table 2. Model performance comparison.

Mode	TP	TN	FP	FN	Recall	Precision	F1 Score	Accuracy Score	Process Time	Memory Usage (GPU)	Efficiency Score	Total Score
PCF-RWKV 7B-Agent	26	28	22	24	0.52	0.54	0.53	26.53	medium	low	35.00	61.53
GPT-4 Base	43	42	8	7	0.86	0.84	0.85	42.57	long	very high	15.00	57.57
PCF-RWKV 7B-Base	17	19	31	33	0.34	0.35	0.35	17.35	short	low	40.00	57.35
RWKV LoRA-LCI	12	8	42	38	0.24	0.22	0.23	11.54	short	low	40.00	51.54
RWKV 7B-Agent	14	13	37	36	0.28	0.27	0.28	13.86	medium	low	35.00	48.86
RWKV LoRA-PCF	7	4	46	43	0.14	0.13	0.14	6.80	short	low	40.00	46.80
RWKV LoRA-Plan	6	5	45	44	0.12	0.12	0.12	5.94	short	low	40.00	45.94
RWKV 7B-Base	6	4	46	44	0.12	0.12	0.12	5.88	short	low	40.00	45.88
qwen1.5 14B-Base	9	3	47	41	0.18	0.16	0.17	8.49	long	medium	25.00	33.49
GLM 6B-Base	4	5	45	46	0.08	0.08	0.08	4.04	very long	low	25.00	29.04

The RWKV-LoRA-LCI, RWKV-LoRA-PCF and RWKV-LoRA-Plan models, based on PCF-RWKV with partial activation, are used for ablation studies to evaluate component importance in the PCF-RWKV model.

Regarding the accuracy score, the PCF-RWKV (7B-Agent) scored 26.53, showing moderate performance with room for improvement, while GPT-4 (base) achieved an accuracy score of 42.57, indicating strong predictive capability in the task. In comparison, the RWKV-LoRA-LCI model scored 11.54, reflecting significant deficiencies in accuracy. The RWKV-LoRA-PCF and RWKV-LoRA-Plan models scored 6.80 and 5.94, respectively, indicating further declines in performance under their configurations. This clearly shows that GPT-4 (base) outperforms the RWKV series models, particularly highlighting the inadequacies of the RWKV-LoRA series in terms of accuracy.

In terms of total score, the PCF-RWKV (7B-Agent) achieved a total score of 61.53, indicating good performance and a favorable balance between efficiency and accuracy. GPT-4 (base) scored 57.57; despite its high accuracy, its lower efficiency resulted in a total score that did not surpass that of PCF-RWKV. The RWKV-LoRA-LCI model had a total score of 51.54, indicating average performance, while RWKV-LoRA-PCF and RWKV-LoRA-Plan scored 46.80 and 45.94, respectively, reflecting their weaker overall capabilities.

From the perspective of ablation studies, the RWKV-LoRA-LCI, RWKV-LoRA-PCF, and RWKV-LoRA-Plan models are based on the RWKV-7B-BASE model and are designed to evaluate the importance of various components in the PCF-RWKV model through partial activation of different modules. When comparing these three models, RWKV-LoRA-LCI stands out with a total score of 51.54, indicating its relative advantage in model performance. Although its recall (0.24) and precision (0.22) metrics are moderate, RWKV-LoRA-LCI shows better performance across various indicators compared to the other two models.

In contrast, RWKV-LoRA-PCF scored 46.80, and RWKV-LoRA-Plan scored 45.94, both of which are lower than RWKV-LoRA-LCI, suggesting that they perform relatively worse across all metrics. The analysis indicates that RWKV-LoRA-LCI has a more significant impact on enhancing model performance in the ablation study, as its inclusion leads to noticeable improvements in overall performance.

Despite RWKV-LoRA-LCI's relatively strong performance in the ablation study, it still has considerable room for improvement compared to high-performing models like PCF-RWKV-7B-Agent and GPT-4-Base. This suggests that even with the introduction of specific modules, the RWKV-LoRA series models have not yet reached optimal performance levels. Future research should focus on further optimizing the activation strategies of these modules to enhance the overall performance and practical application of the models.

4.2. Benchmark Comparison of PCF-RWKV with General Large Language Models

To evaluate the PCF-RWKV model's general-scenario performance and find its strengths and weaknesses, we conducted benchmark comparison experiments between PCF-RWKV and mainstream large models. This helps us understand PCF-RWKV's characteristics in multi-domain knowledge, math, and coding, supporting model optimization and application expansion.

We used multiple authoritative benchmarks from natural language processing and AI fields:

- **Massive Multitask Language Understanding (MMLU)**: Covers 57 domains. It uses five-shot multiple-choice questions to test the model's knowledge and reasoning across different areas [29].
- **HumanEval**: Focuses on code generation. It gives Python(3.7) programming problems and requires zero-shot code generation, evaluating programming logic and syntax [30].
- **Grade School Math 8K (GSM8K)**: Assesses math reasoning with 8000 elementary-to-middle-school-level problems. The model uses eight-shot examples and needs to solve problems through logical reasoning.

- **MATH**: Deals with challenging math problems in algebra, geometry, etc. It provides four-shot examples and requires complex math reasoning.

We followed strict testing procedures. First, we standardized input formats for all models. For text-based questions and code-generation tasks, inputs were made consistent. All test data were pre-processed to remove redundancy and correct errors.

Model outputs were scored by strict metrics. In MMLU, scores were based on answer matching. In HumanEval, code functionality and syntax were checked; in GSM8K and MATH, scores were determined by answer consistency.

Since many large models like GPT-4 are non-open-source and resource-intensive, we used their official benchmark results [31–33]. These results are reliable and serve as references for our comparison. For smaller-parameter models like PCF-RWKV 7B, we conducted local tests on a Supermicro SYS-7048GR-TR tower with four RTX 2080 Ti 22G GPUs and two E5-2680V4 CPUs, in a stable software environment (PyTorch 2.0, Ubuntu 22.04 LTS). Each benchmark was run three times to reduce randomness, and average results were used.

This setup uses official large-model results and conducts accurate local tests for small-parameter models, enabling a comprehensive benchmark comparison between PCF-RWKV and other mainstream large models.

In this benchmark evaluation of different models, an analysis of the data in Table 3, Figures 9 and 10 provides a clear insight into the performance of each model across various task scenarios.

Looking at the overall average scores, Claude3-Opus leads with a score of 81.93%, demonstrating superior comprehensive performance. Following are Gemini 1.0 Ultra (76.43%) and Gemini 1.5 Pro (76.00%). In contrast, smaller parameter models such as PCF-RWKV 7B and LLaMA2-7B have relatively lower average scores. Among these, PCF-RWKV 7B has the highest average score of 19.59%.

In the "MMLU 5-shot" test, which assesses the models' ability to understand and reason across multiple domains, Claude3-Opus scores the highest at 86.80%. PCF-RWKV 7B, however, scores only 23.50%, a significant gap from the leading models. This is primarily because PCF-RWKV is trained specifically for PCF domain knowledge, which affects its performance in general multi-domain knowledge tasks, leading to its poor performance in this test.

Table 3. Benchmark evaluation results of different models.

Model	MMLU 5-Shot	HumanEval 0-Shot	GSM8K 8-Shot	MATH 4-Shot	AVERAGE
Claude3-Opus	86.80%	84.90%	95.00%	61.00%	81.93%
Claude3-Haiku	75.20%	75.90%	88.90%	40.90%	70.23%
Gemini 1.0 Ultra	83.70%	74.40%	94.40%	53.20%	76.43%
Claude3-Sonnet	79.00%	73.00%	92.30%	40.50%	71.20%
Gemini 1.5 Pro	81.90%	71.90%	91.70%	58.50%	76.00%
Gemini 1.0 Pro	71.80%	67.70%	86.50%	32.60%	64.65%
GPT-4	86.40%	67.00%	92.00%	52.90%	74.58%
GPT-3.5	70.00%	48.10%	57.10%	34.10%	52.33%
Qwen 1.5 14B	66.30%	32.30%	61.30%	24.80%	46.18%
PCF-RWKV 7B	23.50%	31.73%	17.83%	5.30%	19.59%
Qwen 1.5 7B	56.70%	24.40%	51.60%	11.60%	36.08%
RWKV 7B	48.30%	13.54%	14.37%	4.30%	20.13%
LLaMA2-7B	46.80%	12.80%	16.70%	3.30%	19.90%
chatGLM2-6B	47.90%	10.87%	32.40%	6.50%	24.42%

The benchmarks of GPT, Claude, and Gemini are from the official published results.

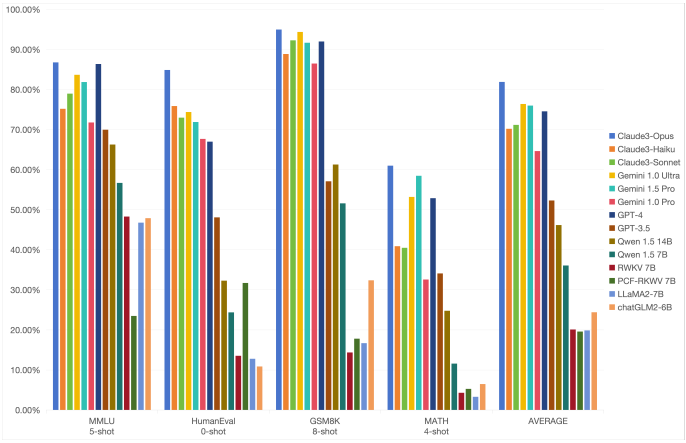


Figure 9. Benchmark evaluation (bar chart). (1) On HumanEval, PCF-RWKV outperforms other small-parameter models. (2) In mathematical benchmarks, PCF-RWKV shows improvements compared to RWKV. (3) PCF-RWKV performance on multi-domain benchmarks like MMLU declines.

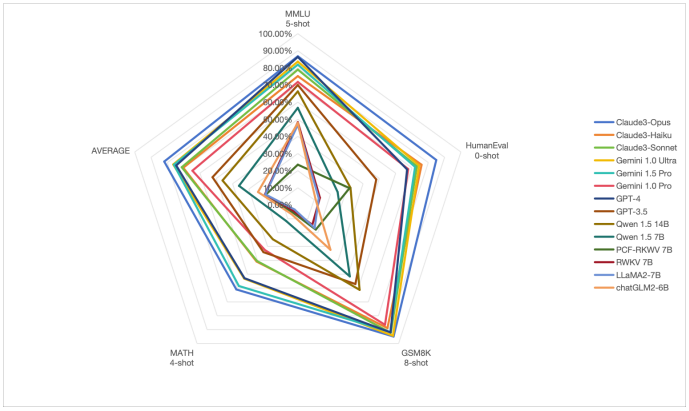


Figure 10. Benchmark evaluation (radar chart). (1) On HumanEval, PCF-RWKV outperforms other small-parameter models. (2) In mathematical benchmarks, PCF-RWKV shows improvements compared to RWKV. (3) PCF-RWKV performance on multi-domain benchmarks like MMLU declines.

For the "HumanEval 0-shot" code generation test, Claude3-Opus again performs well with a score of 84.90%. PCF-RWKV 7B achieves a score of 31.73%, a notable improvement over RWKV 7B's 13.54%. This indicates the optimization of PCF-RWKV in code generation capabilities. Although it still lags behind the top models, it has made significant progress in understanding programming logic and syntax compared to RWKV 7B.

Focusing on the "GSM8K 8-shot" mathematical reasoning test, which covers math problems from elementary to middle school levels, Claude3-Opus scores the highest at 95.00%. PCF-RWKV 7B scores 17.83%, which, while not high, is an improvement over RWKV 7B's 14.37%. This suggests that training for the PCF domain has to some extent enhanced its mathematical reasoning abilities.

In the "MATH 4-shot" test, which focuses on solving advanced math problems, Claude3-Opus scores 61.00%, still at a high level. PCF-RWKV 7B scores only 5.30%, but this is a slight improvement over RWKV 7B's 4.30% in handling complex math problems.

In summary, PCF-RWKV 7B, trained specifically for the PCF domain, falls short in general capabilities compared to mainstream large models. However, it shows a proportional improvement in mathematical and coding abilities compared to RWKV 7B, demonstrating the effectiveness of domain-specific training. Future work could focus on optimizing general capabilities while maintaining domain-specific advantages to enhance its comprehensive performance across multiple scenarios.

5. Discussion

5.1. Comparison with Similar Works in the PCF Assessment Field

Accurately and efficiently estimating the carbon footprint of products is essential for reducing carbon emissions and mitigating climate change. Such efforts are integral to global strategies addressing environmental degradation and global warming. Leveraging large language models (LLMs) for this task offers significant potential due to their advanced capabilities in processing complex data and generating reliable predictions. Deng et al. (2023) proposed an innovative framework, AutoPCF [10], which automates the accounting of product carbon footprints (PCF) using LLMs. This framework was tested across five LLMs to simulate the product lifecycle from cradle-to-gate and generate input–output inventory data. By analyzing the carbon footprints of three case products, the study demonstrated that AutoPCF significantly reduces modeling time from days to minutes, enabling rapid and automated PCF estimation.

To compare with Deng's AutoPCF [10], we conducted the same tests on PCF-RWKV, focusing on its lifecycle inventory (LCI) analysis capabilities and the accuracy of product carbon footprint estimation.

To evaluate the performance of inventory analysis based on large language models (LLMs) compared to expert processes, we adopted the ROUGE method and defined three metrics: precision, recall, and F1 score. Precision is defined as the proportion of correctly retrieved instances among all retrieved instances (where an "instance" refers to an item in the inventory, such as coke, pig iron, oxygen, etc.). Recall is the ratio of correctly retrieved instances to the total number of correct instances. The F1 score, as the harmonic mean of precision and recall, measures the overall performance of the model.

For comparison with the previous work AutoPCF, we selected the same three products as our main test cases: hot-rolled round steel, textile dyeing and printing, and lithium iron phosphate battery production processes and activity inventory generation. By reviewing other literature on the carbon footprint estimation of these three products (hot-rolled round steel [34], dyed fabrics [35,36], lithium iron phosphate battery [37–42]), and based on expert models, we determined the estimated median and uncertainty range to evaluate the error and uncertainty of AutoPCF results.

In this study, we compared the performance of three models—PCF-RWKV, GPT-4, and GLM-6B—in production process modeling and activity inventory generation (with GPT-4 and GLM-6B experimental data sourced from the AutoPCF paper [10]). The results are shown in Figure 11. Here are the detailed comparison results:

In the production process modeling task, PCF-RWKV achieved an average F1 score of 0.60, higher than GPT-4's 0.53 and GLM-6B's 0.37, indicating a better balance between accuracy and recall. Specifically, the PCF-RWKV model achieved a precision of 58%, surpassing GPT-4's 56%, with a recall of 75%, slightly above GPT-4's 72%. This suggests that the PCF-RWKV model is more precise in generating product production processes. In contrast, GLM-6B's precision and recall were 49% and 30%, respectively, significantly lower than the other two models, indicating suboptimal performance in the production process modeling task.

In the activity inventory generation task, the PCF-RWKV model achieved a precision of over 61%, a recall of over 35%, and an F1 score of over 0.41, particularly in dyeing and printing inventory modeling, where the F1 score reached 0.55, slightly higher than GPT-4's 0.50. This result further confirms the superior performance of the PCF-RWKV model in activity inventory generation tasks. In contrast, GLM-6B performed relatively poorly across all three metrics, especially in the hot-rolled round steel activity inventory modeling, with an F1 score of only 0.1, far below both GPT-4 and PCF-RWKV models.

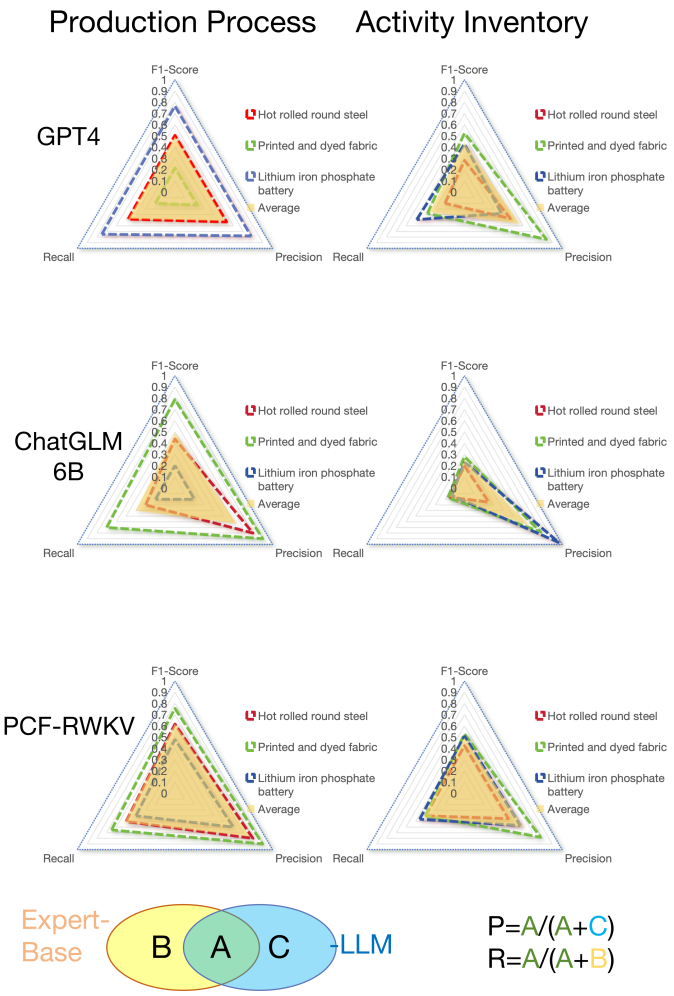


Figure 11. Performance comparison in life cycle inventory analysis. Performance is evaluated by three metrics: Precision (P), Recall (R), and F1 score (F1), ranging from 0 to 1, with 1 being the best and 0 the worst compared to the expert-based process.

The PCF-RWKV model, due to its RNN structure advantage, demonstrates higher resource utilization efficiency and faster processing speed when handling long-sequence data. This characteristic is particularly important on large-scale datasets as it can significantly reduce model training and inference time. While GPT-4 has shown improvements in performance, it may not match the PCF-RWKV model in resource utilization and processing speed. GLM-6B performs better than GPT-4 in resource utilization and processing speed but still falls short of PCF-RWKV.

In product carbon footprint (PCF) accounting, the initial step is to decompose the product lifecycle's production processes to obtain each process's inputs, energy usage, and outputs. The greenhouse gas (GHG) emissions for each process can be calculated using the following formula:

$$E_s = \sum(AD_f \times EF_f) \tag{12}$$

where E_s represents the GHG emissions of process s; AD_f and EF_f are the activity data and the corresponding GHG emission factor of activity f, such as raw material inputs or energy consumption [43].

The estimation effectiveness of the AutoPCF model is measured against the expert-based results by error (Equation (13)) and adjusted coefficient of variation (adjusted CV, Equation (14)):

$$\text{error} = \left| 1 - \frac{PCF_{AutoPCF}}{PCF_{expert-based}} \right| \tag{13}$$

$$\text{adjusted C V} = \frac{\sigma}{\mu} = \frac{\sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}}}{\frac{\sum_{l=1}^{n} x_i}{n}}, \quad Q1 - 1.5 \times IQR \leq x_i \leq Q3 + 1.5 \times IQR \tag{14}$$

where σ is the sample standard deviation, μ is the sample mean, x_i is the i-th observation, $\overline{x}$ is the sample mean, n is the number of samples, $Q1$ is the first quartile, $Q3$ is the third quartile, and $IQR = Q3 - Q1$.

Using the production processes and activity inventories generated by the three models, model-generated activity data (inputs and outputs for each process) were used to estimate the carbon footprints of the three products. Each model's estimation process was repeated 20 times to evaluate the stability of the predictions, and their medians and ranges were compared with expert-based model results. The experimental results are shown in Figure 12, with GPT-4 and GLM6B test data referenced from the paper AutoPCF [10].

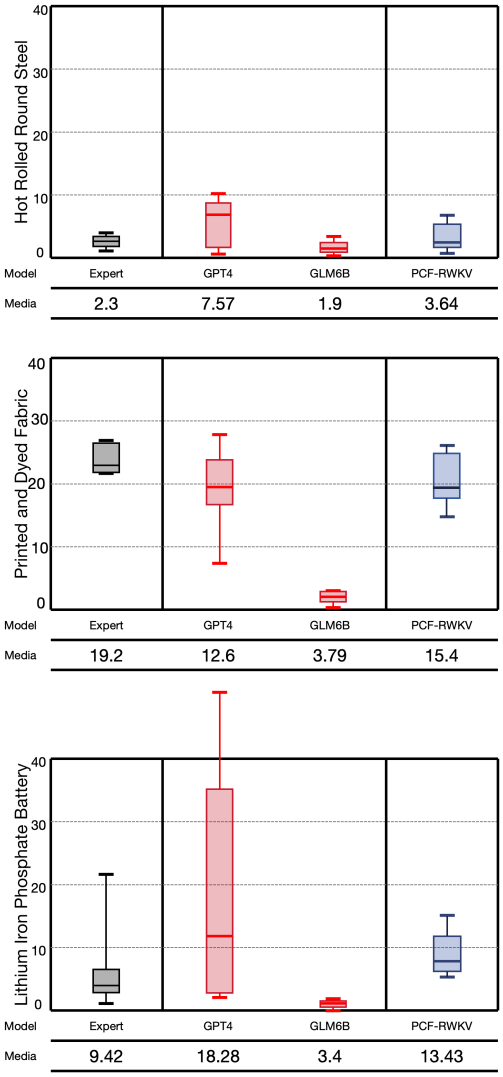


Figure 12. Carbon footprint estimation of selected products. Hot rolled round steel, printed and dyed fabric, and lithium iron phosphate battery.

The experimental results show that the GLM-6B model's average error is 57%, which is relatively low among the three models. The GPT-4 model's average error is 117%; this

higher error might relate to GPT-4's generalization capabilities in specific tasks. Although GPT-4 is a powerful language model, its performance on the specific task of product carbon footprint estimation may not meet expectations due to the quality of training data and model generalization limitations. The GLM-6B model shows better stability, but the carbon footprint data it generates is biased low, also differing from expert predictions. The PCF-RWKV model's average error is 73%, slightly higher than the GLM-6B model.

Although the PCF-RWKV model's average error is slightly higher than the GLM-6B model, its efficiency and stability in handling long-sequence data, along with its linearly growing computational and memory demands, theoretically give it potential advantages over the GLM-6B and GPT-4 models. These characteristics of the RWKV model make it better suited to adapt and scale in practical applications, especially under resource constraints, which is a critical consideration for scalability and cost-effectiveness in real-world applications. Therefore, we can reasonably expect the PCF-RWKV model to perform slightly better than the aforementioned two models in practical applications, particularly when dealing with large-scale and complex product carbon footprint estimation tasks. Despite not having the lowest error, the PCF-RWKV model's overall performance and potential application advantages make it a promising model for further study and application.

5.2. Model Performance Improvement and Loss of Generalizability

We extract the test results of PCF-RWKV before and after training from the Section 4 and create Figures 13 and 14.

As shown in Figure 13, the PCF-RWKV model has demonstrated notable improvement in product carbon footprint (PCF) assessment, attributed to three key factors. First, it employs the RWKV architecture with linear computational complexity, which is highly effective in handling long-sequence data. Second, it uses three task-specialized low-rank adaptation (LoRA) heads for domain-specific training. This reduces data interference and memory waste. Third, it incorporates multi-agent technology, enabling more efficient and accurate PCF accounting.



Figure 13. Comparison of performance and generalization before and after training (1).

However, the PCF-RWKV model's optimizations for product carbon footprint (PCF) assessment have caused a significant loss of generalizability. As shown in Figure 14, its performance on the Massive Multitask Language Understanding (MMLU) benchmark has dropped greatly. Before PCF-related training, RWKV architecture scored 48.3% on MMLU, but after training, PCF-RWKV 7B's score in the MMLU 5-shot test decreased to 23.50%, far lower than Claude3-Opus's 86.80%. This shows that specialized PCF training has seriously affected its performance in broader knowledge domains.

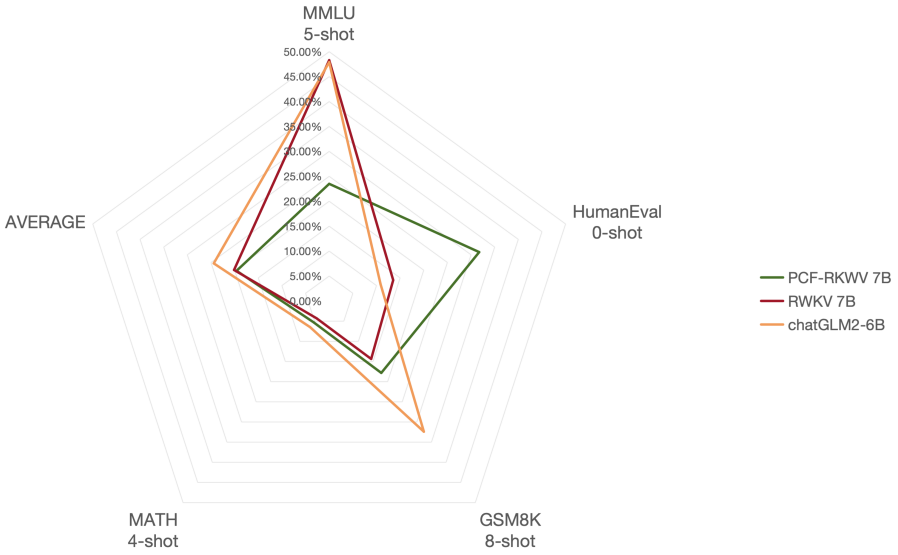


Figure 14. Comparison of performance and generalization before and after training (2).

For small-parameter models, fine-tuning for a specific domain like PCF assessment will impact generalizability. Although we will optimize training datasets and methods, currently, this sacrifice is acceptable to boost the model's PCF-assessment capabilities.

The generalizability problem is rooted in the model's training focus. It highly centers on PCF-related knowledge during training. Intensive training on PCF-specific datasets and using task-specialized LoRA heads make the model overly sensitive to PCF assessment, lacking the ability to answer questions outside this domain, such as those in MMLU's history or the literature parts.

To maintain high PCF-domain precision while solving this issue, future research should focus on enhancing the model's general knowledge.

Data-related improvements are essential. Data augmentation techniques like synthesis and noise injection can diversify training data, and integrating data from related fields into PCF assessment training can provide a more comprehensive knowledge base.

Model structure optimization is also crucial. Regularization methods like L2 regularization and Dropout can prevent overfitting, and a multi-task learning framework can help the model learn PCF assessment and other relevant tasks simultaneously, enhancing its general knowledge and generalization ability.

Multi-modal learning offers new potential. Integrating text, images, and tables allows the model to view PCF assessment from different perspectives, improving both generalization and precision. Cross-modal transfer learning can further expand the model's capabilities.

Model evaluation and optimization are continuous. A dynamic evaluation mechanism is needed to monitor performance in real time and adjust the model promptly. Model compression techniques can reduce complexity and improve generalization, especially in resource-constrained environments.

Through comprehensive exploration of these directions, the PCF-RWKV model can potentially enhance generalization while maintaining high PCF-assessment precision, expanding its application scope and contributing to related research fields.

5.3. Practical Applications and Cross-Industry Adaptability

The PCF-RWKV model holds great potential in practical applications across various industries. In manufacturing, its accurate PCF assessment can help companies identify high-carbon-emission production processes. For example, in automotive manufacturing,

precise carbon footprint calculation for component production allows manufacturers to optimize energy use in metal-smelting or select greener raw materials, reducing emissions and meeting environmental regulations and consumer demands. Precise PCF assessment guides industries to cut emissions, contributing to global climate-change efforts. In the industrial field, with stricter carbon regulations and growing corporate social responsibility, companies using the model to optimize their carbon footprints can gain a market edge. The model's linear computational complexity makes it suitable for large-scale industrial applications. In industries with complex supply chains like electronics, it can handle large-volume data related to component production, transportation, and use, and assess entire product portfolios' carbon footprints. In the energy industry, it can evaluate different energy-generation methods' carbon footprints, supporting the shift to a low-carbon energy system.

However, to fully realize its cross-industry adaptability, the model needs improvement. Integrating data from multiple industries is key. Different industries have distinct carbon-emission patterns despite similar basic calculation principles. Developing industry-specific modules, tailored to each industry's unique features, could be a solution. Also, continuous knowledge transfer and adaptation are essential. As industries evolve, the model should be updated by retraining with the latest relevant data.

6. Conclusions

This paper introduces a novel large-scale model for product carbon footprint assessment, PCF-RWKV, based on the innovative RWKV architecture. The model aims to address challenges faced by traditional lifecycle assessment methods, such as the subjectivity in defining product boundaries and the time-consuming nature of constructing lifecycle inventories. The PCF-RWKV model integrates multiple stacked residual blocks and task-specific LoRA heads and is trained using low-rank adaptive techniques on a product carbon footprint assessment knowledge dataset. With 7 billion parameters, the model can be efficiently deployed on consumer-grade single GPUs, ensuring enterprise data security. By incorporating multi-agent technology, PCF-RWKV can automatically generate models of product production processes, raw material energy usage, and waste emissions, construct lifecycle models, and calculate carbon footprints, providing enterprises with an efficient and secure carbon footprint assessment solution.

The highlights of the PCF-RWKV model lie in its innovative architecture and training methodology, which offer significant advantages in assessment efficiency and resource consumption. Compared to existing technologies, PCF-RWKV demonstrates higher resource utilization efficiency and faster processing speed when handling long sequence data, which is particularly important for large-scale datasets as it significantly reduces model training and inference time. Additionally, the PCF-RWKV model exhibits better balance in product carbon footprint assessment tasks than existing models, especially in terms of recall and F1 scores, indicating its precision in generating product production processes.

Despite the significant advantages of the PCF-RWKV model in evaluation efficiency and resource consumption, this study also has certain limitations. Firstly, the scale and diversity of the training dataset might limit the model's accuracy in specific scenarios. Secondly, the impact of the multi-agent architecture on the model's operational speed requires further research and optimization to ensure responsiveness and efficiency in practical applications.

Future research directions could expand into the following areas: first, increasing the scale and coverage of the dataset to enhance the model's generalization capability and accuracy. Second, exploring the model's applicability across different industries and product types and developing corresponding adjustment strategies to meet the specific needs of

various fields. Additionally, further optimization of agent technology is needed to improve the model's operational efficiency. Finally, considering the model's interpretability and transparency, future work could focus on enhancing the model's explanatory capabilities to help enterprises better understand and trust the assessment results. Through these efforts, the PCF-RWKV model is expected to play a greater role in the field of product carbon footprint assessment, providing robust technical support for enterprises to achieve sustainable development goals.

Author Contributions: Conceptualization, Z.L.; Methodology, Z.L.; Software, Z.L. and P.T.; Validation, P.T. and X.W.; Formal analysis, Z.L., P.T. and X.W.; Investigation, Z.L. and X.W.; Writing—original draft, Z.L. and P.T.; Writing—review and editing, P.M. and X.L.; Supervision, P.M. and X.L.; Funding acquisition, P.M. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: Funded by the National Key Research and Development Program of China through grant number 2022YFC3901701, project title "Environmental Load Calculation Method for the Full Lifecycle of Mobile Phones and Tablets".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in the article.

Acknowledgments: The authors would also like to thank the project team members for their invaluable contributions and collaboration throughout the study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LoRA	Low-Rank Adaptation
LLM	Large Language Model
LCA	Lifecycle Assessment
LCI	Lifecycle Inventory
PCF	Product Carbon Footprint

References

1. Can, W.; Ruoshui, S.; Jiutian, Z. Supportive technologies and roadmap for China's carbon neutrality. *China Econ.* **2021**, *16*, 32–70.
2. Eggleston, H.; Buendia, L.; Miwa, K.; Ngara, T.; Tanabe, K. *2006 IPCC Guidelines for National Greenhouse Gas Inventories*; IPCC: Geneva, Switzerland, 2006.
3. Rüdele, K.; Wolf, M. Identification and Reduction of Product Carbon Footprints: Case Studies from the Austrian Automotive Supplier Industry. *Sustainability* **2023**, *15*, 14911. [CrossRef]
4. Trebbin, A.; Geburt, K. Carbon and Environmental Labelling of Food Products: Insights into the Data on Display. *Sustainability* **2024**, *16*, 10876. [CrossRef]
5. Islam, S.; Ponnambalam, S.; Lam, H.L. Review on life cycle inventory: Methods, examples and applications. *J. Clean. Prod.* **2016**, *136*, 266–278. [CrossRef]
6. Weyand, A.; Bausch, P.; Engel, B.; Metternich, J.; Weigold, M. Analysis of Uncertainty Factors in Part-Specific Greenhouse Gas Accounting. *Sustainability* **2023**, *15*, 16871. [CrossRef]
7. Yang, Y.; Yue, X.; Luo, Y.; Jin, L.; Jia, B. Building Information Modeling–Life Cycle Assessment: A Novel Technology for Rapid Calculation and Analysis System for Life Cycle Carbon Emissions of Bridges. *Sustainability* **2024**, *16*, 10574. [CrossRef]
8. Raiaan, M.A.K.; Mukta, M.S.H.; Fatema, K.; Fahad, N.M.; Sakib, S.; Mim, M.M.J.; Ahmad, J.; Ali, M.E.; Azam, S. A review on large Language Models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access* **2024**, *12*, 26839–26874. [CrossRef]
9. Al-Dokhny, A.; Alismaiel, O.; Youssif, S.; Nasr, N.; Drwish, A.; Samir, A. Can Multimodal Large Language Models Enhance Performance Benefits Among Higher Education Students? An Investigation Based on the Task–Technology Fit Theory and the Artificial Intelligence Device Use Acceptance Model. *Sustainability* **2024**, *16*, 10780. [CrossRef]

10. Deng, Z.; Liu, J.; Luo, B.; Yuan, C.; Yang, Q.; Xiao, L.; Zhou, W.; Liu, Z. AutoPCF: Efficient Product Carbon Footprint Accounting with Large Language Models. *arXiv* **2023**, arXiv:2308.04241.
11. Vauche, L.; Guillemaud, G.; Lopes Barbosa, J.C.; Di Cioccio, L. Cradle-to-Gate Life Cycle Assessment (LCA) of GaN Power Semiconductor Device. *Sustainability* **2024**, *16*, 901. [CrossRef]
12. Gonzàlez-Gonzàlez, J.; Garcìa-Méndez, S.; De Arriba-Pérez, F.; Gonzàlez-Castaño, F.J.; Barba-Seara, o. Explainable Automatic Industrial Carbon Footprint Estimation From Bank Transaction Classification Using Natural Language Processing. *IEEE Access* **2022**, *10*, 126326–126338. [CrossRef]
13. Zeng, Q.; Wan, Y.; Ma, J.; Song, H.; Lin, K. Online Accounting Method for Carbon Emission of Coal Mines owned by the Coal Industry Group. *J. Comput. Electron. Inf. Manag.* **2023**, *11*, 84–92. [CrossRef]
14. Wang, Y.; Luo, D. Design of Carbon Emission Accounting and Monitoring System Based on Artificial Intelligence Algorithm. In Proceedings of the 2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT), Bengaluru, India, 20–21 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–5.
15. Huang, R.; Mao, S. Carbon Footprint Management in Global Supply Chains: A Data-Driven Approach Utilizing Artificial Intelligence Algorithms. *IEEE Access* **2024**, *12*, 89957–89967. [CrossRef]
16. Peng, B.; Alcaide, E.; Anthony, Q.; Albalak, A.; Arcadinho, S.; Biderman, S.; Cao, H.; Cheng, X.; Chung, M.; Grella, M.; et al. Rwkv: Reinventing rnns for the transformer era. *arXiv* **2023**, arXiv:2305.13048.
17. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2023**, arXiv:1706.03762.
18. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
19. Agiza, A.; Neseem, M.; Reda, S. MTLoRA: Low-Rank Adaptation Approach for Efficient Multi-Task Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 16196–16205.
20. Tian, C.; Shi, Z.; Guo, Z.; Li, L.; Xu, C. HydraLoRA: An Asymmetric LoRA Architecture for Efficient Fine-Tuning. *arXiv* **2024**, arXiv:2404.1924.
21. Zhai, S.; Talbott, W.; Srivastava, N.; Huang, C.; Goh, H.; Zhang, R.; Susskind, J. An attention free transformer. *arXiv* **2021**, arXiv:2105.14103.
22. Rashid, K.M.; Louis, J. Times-series data augmentation and deep learning for construction equipment activity recognition. *Adv. Eng. Inform.* **2019**, *42*, 100944. [CrossRef]
23. Bolón-Canedo, V.; Sánchez-Maroño, N.; Alonso-Betanzos, A. A review of feature selection methods on synthetic data. *Knowl. Inf. Syst.* **2013**, *34*, 483–519. [CrossRef]
24. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *arXiv* **2021**, arXiv:2106.09685.
25. Talebirad, Y.; Nadiri, A. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *arXiv* **2023**, arXiv:2306.03314.
26. Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Zhang, S.; Zhu, E.; Li, B.; Jiang, L.; Zhang, X.; Wang, C. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv* **2023**, arXiv:2308.08155.
27. Townsend, J.T. Theoretical analysis of an alphabetic confusion matrix. *Percept. Sychophys.* **1971**, *9*, 40–50. [CrossRef]
28. GLM, T.; Zeng, A.; Xu, B.; Wang, B.; Zhang, C.; Yin, D.; Rojas, D.; Feng, G.; Zhao, H.; Lai, H.; et al. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. *arXiv* **2024**, arXiv:2406.12793.
29. Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; Steinhardt, J. Measuring massive multitask language understanding. *arXiv* **2020**, arXiv:2009.03300.
30. Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H.P.D.O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. Evaluating large language models trained on code. *arXiv* **2021**, arXiv:2107.03374.
31. Anthropic, A. Claude 3.5 sonnet model card addendum. *Claude-3.5 Model Card* **2024**, *3*, 1–8.
32. Team, G.; Georgiev, P.; Lei, V.I.; Burnell, R.; Bai, L.; Gulati, A.; Tanzer, G.; Vincent, D.; Pan, Z.; Wang, S.; et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv* **2024**, arXiv:2403.05530.
33. OpenAI. Hello GPT-4o. 2024. Available online: https://openai.com/index/hello-gpt-4o/ (accessed on 13 May 2024).
34. Khalid, Y.; Wu, M.; Silaen, A.; Martinez, F.; Okosun, T.; Worl, B.; Low, J.; Zhou, C.; Johnson, K.; White, D. Oxygen enrichment combustion to reduce fossil energy consumption and emissions in hot rolling steel production. *J. Clean. Prod.* **2021**, *320*, 128714. [CrossRef]
35. Thomas, B.; Fishwick, M.; Joyce, J.; van Santen, A. *A Carbon Footprint for UK Clothing and Opportunities for Savings—Final Report*; WRAP: London, UK, 2012.
36. Beton, A.; Dias, D.; Farrant, L.; Gibon, T.; Le Guern, Y.; Desaxce, M.; Perwueltz, A.; Boufateh, I.; Wolf, O.; Kougoulis, J.; et al. *Environmental Improvement Potential of Textiles (IMPRO Textiles)*; European Commission: Brussels, Belgium, 2014.
37. Tao, Y.; Sun, T.; Wang, Z. Uncovering various paths for environmentally recycling lithium iron phosphate batteries through life cycle assessment. *J. Clean. Prod.* **2023**, *393*, 136263. [CrossRef]

38. Amarakoon, S.; Smith, J.; Segal, B. *Application of Life-Cycle Assessment to Nanoscale Technology: Lithium-Ion Batteries for Electric Vehicles*; Technical Report; U.S. EPA: Washington, DC, USA, 2013.
39. Majeau-Bettez, G.; Hawkins, T.R.; Strømman, A.H. Life cycle environmental assessment of lithium-ion and nickel metal hydride batteries for plug-in hybrid and battery electric vehicles. *Environ. Sci. Technol.* **2011**, *45*, 4548–4554. [CrossRef] [PubMed]
40. Zackrisson, M.; Avellán, L.; Orlenius, J. Life cycle assessment of lithium-ion batteries for plug-in hybrid electric vehicles–Critical issues. *J. Clean. Prod.* **2010**, *18*, 1519–1529. [CrossRef]
41. Lai, X.; Gu, H.; Chen, Q.; Tang, X.; Zhou, Y.; Gao, F.; Han, X.; Guo, Y.; Bhagat, R.; Zheng, Y. Investigating greenhouse gas emissions and environmental impacts from the production of lithium-ion batteries in China. *J. Clean. Prod.* **2022**, *372*, 133756. [CrossRef]
42. Lu, Q.; Wu, P.F.; Shen, W.X.; Wang, X.C.; Zhang, B.; Wang, C. Life cycle assessment of electric vehicle power battery. In *Proceedings of the Materials Science Forum*; Trans Tech Publications Ltd.: Bäch, Switzerland, 2016; Volume 847, pp. 403–410.
43. Shi, X.; Meier, H. Carbon emission assessment to support planning and operation of low-carbon production systems. *Procedia CIRP* **2012**, *3*, 329–334. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.