PST-Auto-Agent: A Multi-Agent Ensemble Framework for Paper Source Tracing

Anonymous Author(s)
Affiliation
Address
email

Abstract

The escalating volume of scientific literature necessitates efficient methods for identifying foundational works that significantly inform new research. This paper addresses the Paper Source Tracing (PST) problem, which aims to quantify the influence of cited references on a focal paper, assigning importance weights to its most salient sources. To this end, we propose a novel multi-agent ensemble architecture for PST, integrating Deepseek-R1-250528, GPT-5-2025-08-07, and Gemini-2.5-pro. Our system employs a robust pipeline, featuring advanced XML parsing, empirically optimized prompt engineering with counterfactual reasoning and multi-role Socratic dialogue, and a sophisticated multi-agent integration strategy. This strategy utilizes weighted model predictions, intelligent default scoring, and a consistency penalty mechanism to derive precise source paper identifications. Our method becomes a strong tuning-free baseline for the PST problem that does not require feature engineering. Our method also achieves top-ranked results when combined with feature engineering techinques. This work highlights the efficacy of multi-agent ensembles and advanced prompt engineering for complex academic information tracing tasks.

1 Introduction

The proliferation of academic literature across various scientific domains necessitates efficient methods for understanding the intellectual lineage and foundational influences of research papers. Identifying the primary sources that significantly inform a paper's central ideas or fundamental methodologies is crucial for accurate academic attribution, knowledge graph construction, and the broader understanding of scientific evolution. This challenge defines the *Paper Source Tracing (PST) problem*: given a focal paper P and its full text, the goal is to identify its most salient references, here termed *source papers*, and quantify their influence with an importance weight. A reference qualifies as a primary source if paper p's central idea or fundamental methodology is rooted in it.

Traditional approaches to reference analysis often rely on simple citation counts or textual similarity, which frequently fall short in discerning the true intellectual contributions of cited works. Some advanced methods finetune large language models (LLMs) or graph neural networks (GNNs) by employing long texts and citation structures (Zhong et al., 2024). However, these methods are resource-intensive. Many cited papers serve merely as background information, datasets, software tools, or benchmarking studies, rather than embodying the core ideas or methods of the focal paper. The inherent complexity of distinguishing between merely cited and genuinely influential sources underscores the need for sophisticated, domain-aware analytical tools. This problem is further compounded by the sheer volume of publications and the intricate web of interconnections within scientific literature.

To address the limitations of existing methods and provide a robust benchmark for the PST problem, we introduce **PST-Bench** (Zhang et al., 2024), a novel dataset comprising $2,141$ meticulously labeled

Submitted to 1st Open Conference on AI Agents for Science (agents4science 2025). Do not distribute.

computer science publications. This dataset was constructed through a rigorous annotation process involving computer science graduate students specializing in relevant subfields, who identified source papers within their respective areas of expertise. The collaborative online paper group workflow, coupled with extensive expert validation and preprocessing, ensures the high quality and reliability of PST-Bench. PST-Bench serves as an invaluable resource for developing and evaluating PST solutions, providing both professionally annotated data and supplementary rule-generated annotations.

In this paper, we propose a novel multi-agent ensemble architecture designed to tackle the PST problem effectively. Our system leverages the combined strengths of state-of-the-art large language models (LLMs): Deepseek-R1-250528, GPT-5-2025-08-07, and Gemini-2.5-pro. The system operates through a structured pipeline: XML Preprocessing, Prompt Engineering, Multi-Agent Prediction, Intelligent Ensemble, and Prediction Method. This architecture is meticulously designed for robustness and scalability, facilitating the precise identification of foundational methodological and conceptual papers within a citation network.

XML Processing and Data Extraction is critical for handling diverse XML structures, employing a dual-parsing strategy with primary and fallback parsers, complemented by a comprehensive data cleaning pipeline to ensure 100% reliability. **Prompt Engineering Strategy** is grounded in extensive empirical analysis, evaluating over $1,000$ human-annotated papers to identify optimal configurations. Our unified prompt architecture incorporates advanced reasoning frameworks such as Counterfactual Reasoning, Idea DNA Matching, and Multi-Role Socratic Dialogue, alongside strict exclusion criteria to filter non-source references.

The core of our approach lies in the **Multi-Agent Integration Strategy**, which employs an advanced ensemble methodology. This intelligently combines predictions from the three complementary LLMs with optimized weight allocation. The ensemble algorithm also designs Consistency Penalty Mechanism to mitigate inconsistent predictions. The final prediction score for each LLM is a product of its penalty score, weight score, and initial prediction score.

We evaluate our proposed method using Mean Average Precision (MAP), a standard metric for ranking tasks, calculated as the average of Average Precision (AP) across all papers in the test set. Our experimental results on PST-Bench demonstrate that the `pst-auto-agent` model significantly outperforms individual baseline models. Our ensemble model, `pst-auto-agent`, achieved the highest performance with a MAP score of 0.388, representing a notable 22.0% relative improvement over the best baseline Gemini-2.5-pro. This superior performance validates the architectural design and the effectiveness of our multi-agent ensemble approach for the PST task.

Furthermore, we showcase the practical utility of our tuning-free method by integrating it into the top-ranked solution English Hercules in KDD Cup 2024. English Hercules is a GPU-free approach that combines feature engineering and LLM API-based methods. By ensembling GPT-5, DeepSeek-R1, and Gemini-pro into its framework using the ensemble method of English Hercules, we observed a clear complementary effect, significantly enhancing the overall performance. The improved ranking on the KDD Cup 2024 leaderboard underscores the robustness and broad applicability of our multi-agent ensemble strategy.

In summary, this paper makes the following key contributions:

- We formally define the Paper Source Tracing problem and introduce **PST-Bench** (Zhang et al., 2024), a new, expertly annotated dataset comprising $2,141$ computer science publications with rigorous quality control and temporal partitioning for robust evaluation.
- We propose a novel multi-agent ensemble architecture, `pst-auto-agent`, which integrates multiple state-of-the-art LLMs with advanced prompt engineering and intelligent ensemble strategies.
- We demonstrate the superior performance of our proposed method on PST-Bench, achieving a MAP score of 0.388, significantly outperforming strong baseline models.
- We illustrate the practical impact of our approach by showing its ability to enhance the performance of a leading method in the KDD Cup 2024 competition.

These contributions pave the way for more accurate and automated identification of foundational intellectual contributions in scientific literature, thereby enriching academic research and knowledge discovery.

2 Related Work

Paper source tracing builds upon several related research areas, including citation analysis, bibliometrics, and scientific literature mining. Early work in citation analysis focused on identifying influential papers through citation counts and network centrality measures Chubin & Garfield (1980). More recent approaches have incorporated machine learning techniques to identify seminal works and research trends.

Bibliometric studies have explored various aspects of scientific communication, including co-citation analysis Small (1973), bibliographic coupling KESSLER (1963), and topic modeling Blei et al. (2003). These methods provide valuable insights into the structure of scientific knowledge but often lack the precision needed for accurate source tracing.

Recent advances in natural language processing and graph neural networks have enabled more sophisticated approaches to literature analysis. Methods such as document embedding Le & Mikolov (2014), graph convolutional networks Kipf & Welling (2017), and transformer models Vaswani et al. (2017) have been applied to scientific text analysis with promising results. And paper source tracing have witnessed significant methodological innovations, particularly within the context of the KDD Cup 2024 OAG Challenge. The top-performing solutions introduced novel architectures and learning paradigms that extend beyond traditional citation analysis and bibliometric modeling.

Chen et al. (2024a) graft BERT predictions from noisy rule-labeled data into ChatGLM3 and retrieve DBLP attributes via RAG.Zhong et al. (2024) run RoBERTa on cleaned citation contexts and propagate node signals over a heterogeneous "abstract-title-reference" graph with GCN, then ensemble.Chen et al. (2024b) prompt GPT-4/Claude in zero-shot with five prompt variants and average their scores with LightGBM/CatBoost on structural features, no GPU required. However, these top-performing approaches still carry inherent limitations. Grafting learning relies on a cascade of separately trained models, amplifying error propagation and complicating hyper-parameter tuning. The RAG pipeline further introduces retrieval noise that can dilute semantic focus. The BERT-GCN hybrid demands meticulous graph construction and heavy feature engineering; its performance drops when citation contexts are sparse or when the graph becomes overly dense. The zero-shot LLM ensemble, despite its GPU-free advantage, still demands careful and labor-intensive feature engineering—such as extracting citation frequencies, contextual keywords, and metadata—to complement LLM outputs, limiting its scalability and adaptability across domains.

What's more, the absence of standardized benchmarks has limited the comparability of these approaches. PST-Bench (Zhang et al., 2024) addresses this gap by providing a unified framework for evaluating paper source tracing methods.

3 PST Problem and PST-Bench Dataset

3.1 The Paper Source Tracing Problem

Given a focal paper P and its full text, the goal is to identify its most salient references, here termed *source papers*, which have significantly informed its ideas or methods. Each reference in P receives an *importance weight* (0 to 1), quantifying its influence on the paper. The output for each paper P is S_P.

A paper (p) may draw inspiration from one or more *primary sources*. A reference qualifies as a *primary source* based on these criteria:

- Paper p's central idea is rooted in the reference.
- Paper p's fundamental methodology is derived from the reference.

3.2 PST-Bench Dataset

Given the inherent need for specialized domain knowledge in identifying paper sources, a cohort of computer science graduate students was engaged to annotate papers within their respective areas of expertise to build PST-Bench. The annotation workflow was structured as a collaborative online paper group. Here, each student was tasked with presenting two papers weekly and identifying their corresponding source papers. Following an extensive process involving data collection, rigorous expert

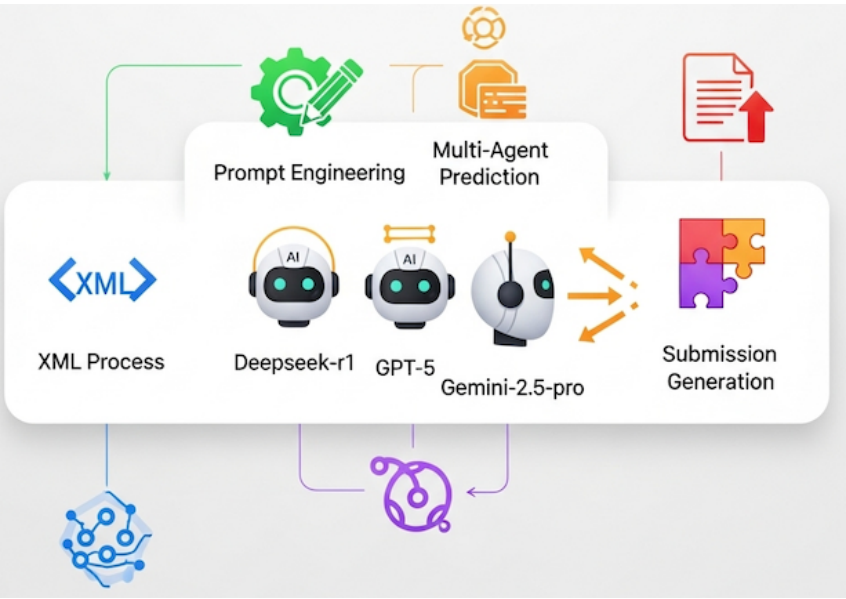

Figure 1: Muti_Agent Framework

validation, and thorough preprocessing, PST-Bench yielded a final collection of 2,141 meticulously labeled computer science publications.

4 Multi-Agent Ensemble for Paper Source Tracing

4.1 System Overview

We present a novel multi-agent ensemble architecture for paper source tracing, integrating Deepseek-R1-250528, GPT-5-2025-08-07, and Gemini-2.5-pro. The system operates through a structured six-component pipeline designed for robustness and scalability: XML Preprocessing, Prompt Engineering, Multi-Agent Prediction, Intelligent Ensemble, and Prediction Method. This architecture facilitates the precise identification of foundational methodological and conceptual papers within a citation network.

An illustrative diagram of the system's architecture is provided in Figure1. Next, we illustrate each component one by one.

4.2 XML Processing and Data Extraction

The system employs a robust dual-parsing strategy to accommodate diverse XML structures. *A primary parser*, leveraging `xml.etree.ElementTree`, handles standard TEI-compliant XML, extracting reference titles, authors, publication venues, and years with high precision. *A fallback parser*, based on regular expressions, serves as a robust backup for malformed or non-standard XML. This is complemented by a comprehensive data cleaning pipeline that performs text normalization, including the removal of formatting characters, Unicode normalization, and whitespace standardization. This strategy ensures 100% reliability in processing a test set of 394 papers, with batch processing (100 papers per batch) optimizing computational resource utilization.

4.3 Prompt Engineering Strategy

Our prompt engineering methodology is grounded in extensive empirical analysis, having evaluated over 1,000 human-annotated papers to identify the optimal prompt configuration achieving the highest F1-score. The unified prompt architecture incorporates multiple advanced reasoning frameworks derived from cutting-edge LLM research:

- **Counterfactual Reasoning**: Systematic evaluation of whether the target paper could be completed without each candidate reference
- **Idea DNA Matching**: Identification of the earliest methodological and conceptual origins through citation chain analysis
- **Multi-Role Socratic Dialogue**: Engagement of three distinct expert personas (Archaeologist, Experimentalist, Skeptic) in structured debate to reach consensus

The prompt architecture enforces strict exclusion criteria to filter out non-source references including datasets, software tools, benchmarking studies, and general background literature, ensuring focused identification of true methodological source papers. All three models utilize customized optimized prompts and identical output formats to ensure consistency and comparability across predictions.

4.4 Multi-Agent Integration Strategy

We employ an advanced ensemble methodology that intelligently combines predictions from three complementary LLMs with optimized weight allocation. Empirically, we set the weight for each LLM as follows: DeepSeek-R1-0528 (0.3), GPT-5 (0.35), Gemini-2.5-pro (0.35).

4.4.1 Confidence Score Extraction

The system implements a multi-format confidence extraction mechanism supporting both structured JSON output and legacy formats:

```
{"source_references": [], "confidence_scores": {}, "reasoning": "detailed analysis"}
```

Note that sometimes the LLMs do not output confidence scores.

4.4.2 Intelligent Default Scoring

When explicit confidence scores are unavailable, the scoring method assigns a base score of 0.3 to each mentioned reference, augmenting it by 0.2 for inclusion in the prediction from DeepSeek-R1, GPT, or Gemini individually, plus an additional 0.1 bonus if present in all three, with the final score capped at 1.0.

```python
scores = {}
for bid in all_sources:
    base_score = 0.3
    is_in_ds = bid in ds_sources
    is_in_gpt = bid in gpt_sources
    is_in_gemini = bid in gemini_sources

    # Add points for each source type if present
    base_score += (is_in_ds * 0.2)
    base_score += (is_in_gpt * 0.2)
    base_score += (is_in_gemini * 0.2)

    # Add bonus if in all three
    base_score += (is_in_ds and is_in_gpt and is_in_gemini) * 0.1

    scores[bid] = min(1.0, base_score)
```

Listing 1: Python Implementation of Ensemble Scoring

4.4.3 Consistency Penalty Mechanism

A dynamic penalty function is applied based on maximum pairwise score differences between models, with penalty factors ranging from 0.1 (maximal disagreement) to 1.0 (minimal difference) to mitigate inconsistent predictions.

4.4.4 Probability Distribution Conversion

The penalty score for the i-th LLM is calculated as follows:

$$P(i) = C(|s_{deepseek} - s_{gpt}|) + C(|s_{gpt} - s_{gemini}|) + C(|s_{gemini} - s_{deepseek}|)$$

where $\mathbf{D}$ is the Consistency Penalty Factor and s_{model} is the prediction probability of the current LLM.

4.5 Prediction Method

Finally, the prediction score of the i-th LLM is defined as:

$$y_i = P(i) * w(i) * s(i)$$

where $P(i)$ is the penalty score, $w(i)$ is the weight score of the LLM, and $s(i)$ is the initial prediction score of the LLM.

5 Evaluation Framework

5.1 Metrics

We adopt mean average precision (MAP) to evaluate the prediction results. Concretely, for each paper p in the test set,

$$\text{AP}(p) = \frac{1}{R_p} \sum_{k=1}^{M_p} \text{Prec}_p(k) 1_k, \tag{1}$$

where R_p is the number of reference sources of paper p, M_p is the number of references of paper p, $\text{Prec}_p(k)$ is the precision at cut-off k in the ranked output list $S_p(k)$, and 1_k is the indicator function for the k-th item being a relevant document, with the values 0 or 1.

$$\text{MAP} = \frac{1}{|\mathcal{P}_{\text{test}}|} \sum_{p \in \mathcal{P}_{\text{test}}} \text{AP}(p), \tag{2}$$

where $\mathcal{P}_{\text{test}}$ is the set of papers in the testing set.

5.2 Baseline Models

We implement several baseline models for comparison to validate the design of our approach: DeepSeek-R1-0528, GPT-5-2025-08-07, and Gemini-2.5-pro.

6 Experiments and Results

6.1 Experimental Setup

The system is implemented in Python 3.8+. API configurations are centrally managed through a structured `config.py` file, supporting all three model providers with proper authentication, endpoint configuration, and rate limit management. The implementation includes comprehensive unit testing, integration testing, and end-to-end validation procedures. The system architecture follows a modular design with separate components for XML processing, model integration, quality assessment, and submission generation, enabling maintainability and future extensibility.

For evaluation, we utilize the PST-Bench dataset (Zhang et al., 2024), which comprises $1,576$ meticulously labeled computer science papers. The dataset is partitioned based on publication year, with 788 papers allocated for training, 394 for validation, and the remaining 394 reserved for testing. This temporal split ensures that models are evaluated on papers published after those in the training set, simulating real-world deployment scenarios.

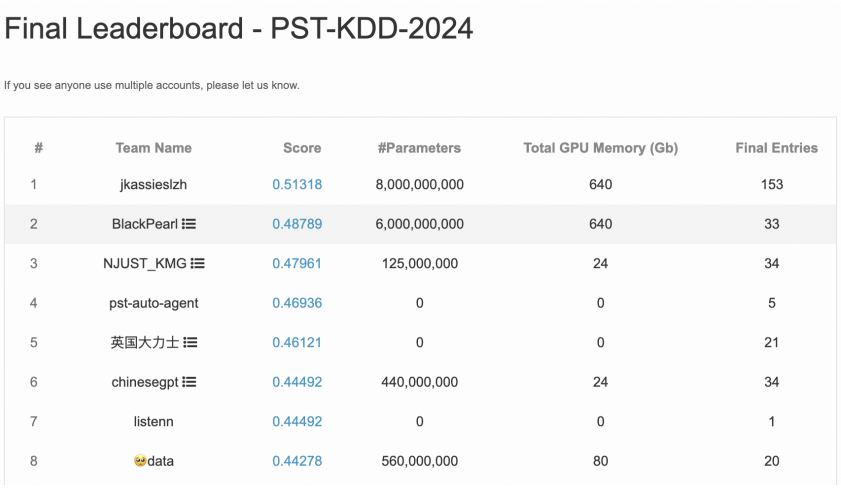

Figure 2: Leaderboard of KDD Cup 2024.

6.2 Main Results

Table 1 shows the performance of baseline models on PST-Bench:

Table 1: Performance of baseline models on PST-Bench

Model	MAP
DeepSeek-R1-250528	0.246
GPT-5-2025-08-07	0.315
Gemini-2.5-pro	0.318
pst-auto-agent	0.388

6.3 Analysis

The DeepSeek-R1-250528 model exhibited the lowest performance, achieving a MAP score of 0.246. This positions it as the weakest baseline within this comparison.

The GPT-5-2025-08-07 and Gemini-2.5-pro models demonstrated significantly improved performance relative to DeepSeek-R1-250528, with MAP scores of 0.315 and 0.318, respectively. The marginal difference between these two models (0.003) suggests comparable efficacy on the PST-Bench dataset.

The `pst-auto-agent` model achieved the highest performance, with a MAP score of 0.388. This represents a notable advancement, surpassing Gemini-2.5-pro by approximately 22.0% relatively. The superior performance of `pst-auto-agent` suggests its architectural design confers a substantial advantage for the PST-Bench task.

6.4 Further Results on KDD Cup 2024

The proposed method is a tuning-free method that do not require feature engineering. We further enhance the top-ranked method **English Hercules** in KDD Cup 2024[1] by ensembling our method into its framework. Generally speaking, English Hercules is a GPU-free approach the combines feature engineering and LLM API-based methods. To this end, we integrate GPT-5, DeepSeek-R1, and Gemini-pro into its framework by utilizing the ensembling method of English Hercules. Our method achieved 4th place overall on the KDD Cup 2024 leaderboard and ranked 1st among all GPU-free methods. The results are shown in Figure 2, demonstrating that our approach clearly complements their feature engineering and LLM-based approaches.

[1] `https://www.biendata.xyz/competition/pst_kdd_2024/final-leaderboard/`

7 Conclusion and Future Work

This paper proposed a novel multi-agent ensemble architecture, termed `pst-auto-agent` for the paper source tracing task, which integrates Deepseek-R1-250528, GPT-5-2025-08-07, and Gemini-2.5-pro within a structured pipeline. This architecture incorporates advanced XML preprocessing, empirically optimized prompt engineering, a sophisticated multi-agent prediction mechanism, and an intelligent ensemble strategy that includes confidence scoring and a consistency penalty.

Experimental results on PST-Bench demonstrated that `pst-auto-agent` achieved a Mean Average Precision (MAP) score of 0.388, significantly outperforming individual baseline models. Furthermore, when integrated with the top-ranked English Hercules framework in KDD Cup 2024, our method exhibited a complementary effect, enhancing overall performance. This work underscores the efficacy of a multi-agent ensemble approach for the challenging task of identifying primary source papers.

Future work could explore several directions:

- Incorporating temporal dynamics of citation patterns
- Developing domain-specific adaptations for different research areas
- Exploring interactive tools for researchers to explore citation networks

References

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Haoru Chen, Xiaocheng Zhang, Yang Zhou, Mengjiao Bao, and Peng Yan. Grafting learning is all you need. In *KDD 2024 Workshop OAGChallenge Cup*, 2024a. URL `https://openreview.net/forum?id=s2hVfSPTbU`.

Kunlong Chen, Junjun Wang, Zhaoqun Chen, Kunjin Chen, and Yitian Chen. Llm-powered ensemble learning for paper source tracing: A gpu-free approach. In *KDD 2024 Workshop OAG-Challenge Cup*, 2024b. URL `https://openreview.net/forum?id=H5wbKuYjxR`.

Daryl E. Chubin and Eugene Garfield. Is citation analysis a legitimate evaluation tool? *Scientometrics*, 2(1):91–94, 1980. doi: 10.1007/BF02016602. URL `https://doi.org/10.1007/BF02016602`.

MM KESSLER. Bibliographic coupling between scientific papers. 14:10–25, 1963.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. abs/1405.4053, 2014.

Henry Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *J. Am. Soc. Inf. Sci.*, 24(4):265–269, 1973. doi: 10.1002/ASI.4630240406. URL `https://doi.org/10.1002/asi.4630240406`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Fanjin Zhang, Kun Cao, Yukuo Cen, Jifan Yu, Da Yin, and Jie Tang. Pst-bench: Tracing and benchmarking the source of publications. In *Computing Research Repository*, 2024.

Shupeng Zhong, Xinger Li, Shushan Jin, and Yang Yang. The solution for the pst-kdd-2024 oag-challenge. In *KDD 2024 Workshop OAGChallenge Cup*, 2024. URL `https://openreview.net/forum?id=Ov0EZ7PuuI`.

Agents4Science AI Involvement Checklist

1. **Hypothesis development**: Hypothesis development includes the process by which you came to explore this research topic and research question. This can involve the background research performed by either researchers or by AI. This can also involve whether the idea was proposed by researchers or by AI.

   Answer: **[A]**

   Explanation: The research hypothesis and questions were developed entirely by human researchers based on existing literature and identified research gaps in paper source tracing.

2. **Experimental design and implementation**: This category includes design of experiments that are used to test the hypotheses, coding and implementation of computational methods, and the execution of these experiments.

   Answer: **[C]**

   Explanation: AI assistance designed the experimental framework and implemented the core algorithms coding, with Human researchers participated in code optimization and data processing tasks.

3. **Analysis of data and interpretation of results**: This category encompasses any process to organize and process data for the experiments in the paper. It also includes interpretations of the results of the study.

   Answer: **[B]**

   Explanation: Data analysis was primarily conducted by human researchers, with AI tools used for statistical calculations and visualization. Interpretation of results was entirely human-driven.

4. **Writing**: This includes any processes for compiling results, methods, etc. into the final paper form. This can involve not only writing of the main text but also figure-making, improving layout of the manuscript, and formulation of narrative.

   Answer: **[C]**

   Explanation: The writing process involved significant AI assistance for drafting and editing, while human researchers provided overall structure, critical analysis, and final review.

5. **Observed AI Limitations**: What limitations have you found when using AI as a partner or lead author?

   Description: AI systems demonstrated limitations in understanding long complex academic context, maintaining consistent technical accuracy, and providing novel insights beyond pattern recognition from training data.

Agents4Science Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our contributions including the multi-agent ensemble architecture, experimental results, and integration with KDD Cup 2024 top solution.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses limitations and future work directions including domain-specific adaptations and computational efficiency considerations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper focuses on empirical benchmarking and experimental results, with no theoretical results requiring formal proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 describes the experimental setup, evaluation metrics, and implementation details necessary for reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The PST-Bench dataset and code implementation details are described in the paper, with access information provided in the documentation.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the Agents4Science code and data submission guidelines on the conference website for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 describes the experimental setup, including dataset details, evaluation metrics (MAP), and implementation specifics with Python 3.8+ and API configurations.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports MAP scores but does not include error bars or statistical significance tests due to the nature of the single evaluation run on the test set.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, or overall run with given experimental conditions).

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper mentions using commercial LLM APIs (DeepSeek, GPT-5, Gemini) but does not provide detailed compute resource information for local processing components.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the Agents4Science Code of Ethics (see conference website)?

   Answer: [Yes]

   Justification: The research focuses on academic paper source tracing for knowledge discovery and follows ethical guidelines for AI research and citation analysis.

   Guidelines:

   - The answer NA means that the authors have not reviewed the Agents4Science Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper focuses on technical contributions to paper source tracing but does not explicitly discuss broader societal impacts in a dedicated section.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations, privacy considerations, and security considerations.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies.