Assessing Deanonymization Risks with Stylometry-Assisted LLM Agent

Anonymous ACL submission

Abstract

The rapid advancement of large language models (LLMs) has enabled powerful authorship inference capabilities, raising growing concerns about unintended deanonymization risks in textual data such as news articles. In this work, we introduce an LLM agent designed to evaluate and mitigate such risks through a structured, interpretable pipeline. Central to our framework is the proposed *SALA* (Stylometry-Assisted LLM Analysis) method, which integrates quantitative stylometric features with LLM reasoning to achieve robust, transparent authorship attribution. Experiments on large-scale news datasets demonstrate that *SALA*, particularly when augmented with a database module, achieves high inference accuracy in various scenarios. Finally, we propose a guided recomposition strategy that leverages the agent's reasoning trace to generate rewriting prompts, effectively reducing authorship identifiability while preserving textual meaning. Our findings highlight both the deanonymization potential of LLM agents and the importance of interpretable, proactive defenses to safeguard author privacy.

1 Introduction

Recent advancements in large language models (LLMs) have enabled the development of increasingly capable AI agents that are already deployed across multiple domains, including research assistance, software engineering, and content moderation (Xi et al., 2023; Ji et al., 2024). However, as LLM agents grow more powerful, their autonomy and reach raise serious safety and ethical concerns. Tasks such as authorship inference, which traditionally required specialized expertise and extensive manual efforts, have become significantly more accessible.

This shift introduces a new privacy risk: the unintended deanonymization of textual data. Individuals such as journalists, whistleblowers, and researchers often rely on anonymity or pseudonymity, and the ability to infer authorship from text can undermine these protections. While traditional stylometric methods have long been studied (Narayanan et al., 2012; Bauersfeld et al., 2023), recent work shows that LLMs can implicitly capture authorial style and perform attribution in a zero-shot setting (Huang et al., 2024). When combined with tool use and iterative reasoning, LLM agents can further amplify these capabilities.

Motivated by this risk, we introduce an LLM agent framework that both evaluates and mitigates deanonymization in text articles, such as printed news. The agent operates through a four-stage pipeline: information extraction, candidate search, candidate matching, and result reflection. At the core of this pipeline, we propose a novel matching strategy SALA (Stylometry-Assisted LLM Analysis), which integrates quantitative stylometric features with LLM reasoning to enable accurate and interpretable authorship attribution. To improve scalability, we incorporate a database module that aggregates author profiles and writing samples.

Beyond risk assessment, we also develop a mitigation strategy. We introduce a guided recomposition method that leverages the agent's reasoning trace to generate targeted rewriting prompts, reducing identifiable stylistic signatures while preserving content. For example, if attribution relies on consistent lexical patterns or sentence structure, the agent can recommend modifications such as lexical diversification or structural variation. In this way, the agent serves as both an adversary and advisor, quantifying authorship risk and guiding authors toward safer textual representations. Together, these components establish a unified framework for understanding and mitigating LLM-agent-driven deanonymization risks.

2 Background

2.1 LLM Agent

LLM agents are typically composed of four key components: *core*, *planning*, *tools*, and *memory* (Liu et al., 2023a; Ruan et al., 2024). These components extend LLMs' capabilities beyond simple text generation to executable actions in both the digital and physical worlds. While these extended capabilities have enabled promising applications across diverse domains, they have also raised growing concerns regarding potential risks and unintended misuse (Gu et al., 2024; Ruan et al., 2024; Zhan et al., 2024; Yang et al., 2024; Mo et al., 2024; Debenedetti et al., 2024).

2.2 Deanonymization

Authorship attribution is the task of determining the most likely author of a text from a candidate set. More broadly, authorship analysis also includes author profiling (inferring demographic or stylistic attributes) and authorship verification (deciding whether a text was written by a given author) (He et al., 2024). Modern approaches leveraging machine-learning-based methods have also achieved success (Narayanan et al., 2012; Bauersfeld et al., 2023). Recently, the introduction of powerful LLMs presents new vectors where the risk of unintentional deanonymization can arise (Huang et al., 2024). A similar analysis has also been used to distinguish LLM-generated from human-composed texts (Mindner et al., 2023; Opara, 2024; Kumarage et al., 2023).

3 Authorship Agent Design

Our proposed deanonymization risk assessment agent is designed as a modular system that integrates LLM reasoning capabilities with external tools for information retrieval, stylometric analysis, and interpretability. The agent operates through four sequential stages: *Information Extraction*, *Candidate Search*, *Candidate Matching*, and *Result Reflection*. Additionally, the agent integrates a *Database Module* to improve inference performance and a *Anonymization Enhancement Module* to mitigate the discovered risk.

Formally, let x denote an anonymous target article, and let $\mathcal{A} = \{a_1, \ldots, a_N\}$ denote a set of candidate authors. The goal is to estimate, for each $a_i \in \mathcal{A}$, whether a_i is a plausible author of x, and to identify the stylistic signals that most strongly support that inference. In the targeted setting, the system evaluates a specific candidate author or a small candidate set. In the open-world setting, the system first retrieves candidates and then ranks them according to their authorship likelihood. The detailed prompt templates for all stages are provided in Appendix A.

3.1 Stage I: Information Extraction

The first stage extracts structured metadata from the target article to guide downstream retrieval. Given an input article x, the agent identifies metadata fields that are informative for narrowing the candidate space, including publication date, topic category, geographic focus, and potential publisher origin. The purpose here is not to determine authorship directly, but to constrain the retrieval space and improve the precision of candidate search.

In the default implementation, this step is performed with an LLM prompt that converts the raw article into a structured record $m(x)$:$\{\texttt{topic}, \texttt{date}, \texttt{location}, \texttt{publisher_cues}\}$. These metadata are then used as soft constraints in the subsequent search stage. By explicitly separating information extraction from matching, the framework makes clear which contextual cues are used for retrieval and which stylometric cues are used for attribution.

3.2 Stage II: Candidate Search

The second stage retrieves a set of plausible author candidates together with representative writing samples. Starting from the metadata record $m(x)$, the system issues search queries over public sources to identify author–article associations that are consistent with the extracted topic, date range, publication context, and geographic information. The output of this stage is a candidate set

$$\mathcal{C}(x) = \{(a_i, D_i)\}_{i=1}^{K}, \tag{1}$$

where a_i is a candidate author and $D_i = \{d_{i1}, \ldots, d_{iM}\}$ is a set of retrieved reference documents written by a_i.

Users specify the maximum number of candidates K and the maximum number of writing samples M per candidate. Candidate retrieval is semi-guided: the LLM is instructed to prioritize authors whose publicly available writings are consistent with the extracted metadata, but retrieval itself is grounded in external evidence rather than closed-book inference. This design reduces the chance that later matching relies solely on the model's prior knowledge.

3.3 Stage III: Candidate Matching

Once the candidate set is established, the agent compares the target article against each candidate's writing profile. We propose *Stylometry-Assisted LLM Analysis* (SALA), a matching procedure that combines explicit stylometric features with LLM-based comparative reasoning.

Stylometric feature representation. For the target article x and each reference article d_{ij}, we extract a set of stylometric features spanning five categories: lexical, syntactic, readability, semantic, and overall style. These features are summarized in Table 1. Let $\phi(\cdot)$ denote the feature extractor. The target article is then represented as $\phi(x)$ and each reference article as $\phi(d_{ij})$. To construct a candidate-level profile, we aggregate features across the retrieved documents for author a_i:

$$P_i = \texttt{Aggregate}\left(\{\phi(d_{ij})\}_{j=1}^{M}\right). \tag{2}$$

For numeric features, aggregation consists of summary statistics such as mean and dispersion across documents. For semantic and style-oriented descriptors, the system forms a compact author profile from the collection of retrieved samples. This aggregation is important because authorship should be inferred from recurring stylistic tendencies rather than from any single reference article.

SALA scoring. Given a target article x and a candidate profile P_i, SALA prompts the LLM to compare the target and candidate using both quantitative feature summaries and representative textual evidence. The LLM is instructed to reason over the structured evidence and return:

1. a scalar authorship score $s(x, a_i) \in [0, 1]$,
2. a short rationale identifying the most influential matching and mismatching features,
3. an uncertainty-aware judgment of whether the candidate is a plausible author.

The final ranking is obtained by sorting candidates according to $s(x, a_i)$. In the targeted setting, the score can also be thresholded to produce a binary attribution decision. In the open-world setting, we report the rank of the true author among all retrieved candidates.

Algorithm algorithm 1 summarizes the full SALA matching procedure. This formulation makes explicit the inputs, intermediate representations, and outputs used by the attribution agent.

Algorithm 1: SALA-based authorship attribution

Input: Anonymous target article x; maximum number of candidates K; maximum number of reference articles per candidate M
Output: Ranked candidate list with authorship scores and feature-based rationales

```
Extract metadata record m(x) from the target article;
Retrieve candidate set C(x) = {(a_i, D_i)}_{i=1}^K using
  metadata-guided search;
Compute target stylometric representation φ(x);
foreach (a_i, D_i) ∈ C(x) do
    foreach d_ij ∈ D_i do
        Compute stylometric representation φ(d_ij);
    Aggregate {φ(d_ij)}_{j=1}^M into candidate profile P_i;

    Prompt the LLM with φ(x), P_i, and
      representative excerpts from D_i;
    Obtain authorship score s(x, a_i) and rationale
      r(x, a_i);
Rank candidates by descending score s(x, a_i);
Return ranked candidates together with rationales
  {r(x, a_i)};
```

Relative to purely qualitative prompting, SALA constrains the LLM to reason over explicit stylometric evidence. Relative to purely feature-based matching, it preserves interpretability by allowing the model to synthesize heterogeneous signals into a coherent authorship judgment.

3.4 Stage IV: Result Reflection

The final stage interprets the matching results to assess deanonymization risk and identify which features of the target article are most disclosing. Rather than treating the highest-scoring author as the only output, the agent examines the comparative evidence produced during matching and summarizes the main reasons that attribution was possible.

Concretely, let $\mathcal{R}(x)$ denote the ranked candidate list and associated rationales. The reflection stage produces a structured explanation $\Gamma(x)$ that includes `risk_level`, `top_signals`, `counterevidence`, and `uncertainty`. The `top_signals` field identifies which cues contributed most strongly to attribution. This step is crucial because it transforms the agent from a purely diagnostic tool into a proactive anonymization assistant.

3.5 Module: Database

The candidate search and matching stages are the most resource-intensive parts of the pipeline because they require repeated retrieval of candidate

Table 1: Stylometric features used in constructing our Stylometry-Assisted LLM Analysis (SALA).

Category	Feature	Description
Lexical	Unique word count	Total number of distinct words used in the text.
	Average word length	Mean number of characters per word.
	Type-token ratio	TTR: ratio of unique words to total words.
	Hapax Legomenon ratio	Proportion of words that appear only once in the text.
Syntactic	Average sentence length	Mean number of words per sentence.
	Stopword count	Number of common functional words (e.g., "the," "is") used.
	Punctuation count	Total occurrences of punctuation marks.
	POS variation count	Diversity of parts of speech (e.g., nouns, verbs, adjectives) used.
Readability	Flesch-reading-ease score	A readability score based on sentence length and syllable count.
Semantic	Polarity	Measuring whether overall tone is positive or negative.
	Subjectivity	Degree to which the text is subjective opinion or objective facts.
Style	Writing style summary	Qualitative style analysis, e.g., formal, casual, with humor/sarcasm, etc.

authors and repeated collection of writing samples. We therefore introduce a database module that caches previously retrieved authors, documents, and stylometric profiles.

The database supports two functions. First, it accelerates candidate search by allowing the system to retrieve previously seen candidates through embedding similarity and keyword matching, invoking external web search only when suitable candidates are not already available. Second, it improves matching stability by storing aggregated author profiles rather than relying only on raw individual documents at inference time.

After an initial warm-up phase, the database contains tuples of the form (a_i, D_i, P_i), where a_i is an author, D_i is the associated document set, and P_i is the aggregated stylometric profile. During inference, the system first checks the database for reusable candidates and profiles before querying external sources. This reduces cost and latency while also making profile construction more consistent across examples.

3.6 Module: Anonymization Enhancement

Building upon the result reflection stage, the *Anonymization Enhancer Module* translates attribution insights into actionable privacy guidance.

Given the reflection output $\Gamma(x)$, the system identifies a subset of highly disclosing features and generates *guided recompose* instructions. These instructions focus on modifying stylistic properties strongly associated with the inferred author while avoiding unnecessary changes to factual content.

Overall, the modular design makes the framework interpretable at each stage: information extraction constrains retrieval, candidate search defines the comparison set, SALA performs evidence-grounded matching, result reflection identifies disclosing signals, and anonymization enhancement uses those signals to support privacy-preserving rewriting.

3.7 Implementation

We implement the agent using the *gpt-4.1-2025-04-14* model (GPT) via the OpenAI Chat Completions API, integrated within the *LangChain* agent framework (LangChain). The stylometric feature extraction tools are developed in Python, utilizing open-source NLP libraries such as *NLTK* [1] for computing lexical, syntactic, and POS-based features. A *web search* function is also implemented following ChatGPT's API specification, enabling real-time retrieval of external information when required. For the following evaluations, we experiment and report results after five repetitions, with the average values reported.

4 Deanonymization Risk Evaluation

4.1 Attack Scenarios

We evaluate the agent in two attack scenarios that reflect different levels of adversarial knowledge.

Targeted Attack. In the *targeted attack*, the adversary tests whether a given article is written by one or more specific candidate authors. Since the candidates are known, the search stage is bypassed and the agent directly performs matching. The test set contains an equal number of articles from target and non-target authors. For multi-target settings, the agent must identify the *exact* author of each article rather than simply detecting membership

[1] https://www.nltk.org/api/nltk.html

Table 2: Percentages of test cases with true authors in the selected potential candidate lists in zero-shot and database augmented (DB-AUG) setting.

Potential Candidates	Zero-Shot	DB-AUG
20	1.2%	78.1%
50	9.8%	89.5%
100	21.1%	93.1%

in the candidate set, making the task increasingly difficult as the number of targets grows.

Open-World Attack. In the *open-world attack*, the adversary has no prior knowledge of the author. The agent must first retrieve candidate authors and then rank them by likelihood. Given a test set of articles from diverse authors, performance is evaluated based on whether the true author appears among the top-ranked predictions.

4.2 Evaluation Settings

Metrics. Performance is evaluated separately for each attack scenario. For *Targeted Attack*, we report **F1-score** to capture both precision and recall when determining whether an article is authored by a specific candidate. For *Open-World Attack*, we report **Top-1** and **Top-3 accuracy**, measuring whether the true author appears among the top-ranked predictions returned by the agent.

Datasets. We conduct experiments on two large-scale news datasets: *All the News 2.0*[2] and *CrossNews* (Ma et al., 2025). *All the News 2.0* contains over 2.5 million articles from hundreds of U.S. media outlets, covering diverse topics, time periods, and writing styles. It includes byline information for more than 13,000 authors, enabling large-scale author-level analysis. *CrossNews* complements this dataset with richer cross-domain coverage, including multiple writing modalities from the same authors (e.g., news articles, blog posts, and social media content), allowing us to examine stylistic variation across contexts. To assess potential data contamination and memorization effects in LLMs, we conduct additional controlled experiments with details presented in the Appendix B. Besides using news articles, we also experiment with academic writings where anonymity is critical (e.g., peer review), see the Appendix C for more details.

4.3 Candidate Search Results

Zero-Shot Search. Given the challenging scenario of open-world authorship attribution, the agent must first narrow down potential candidates. Following the pipeline introduced in section 3, after obtaining the article, the agent utilizes both the web search function and the LLM's internal knowledge to generate a list of potential candidates. To ensure efficient downstream analysis, we set the default to 20 candidates. We select a random subset of 1,000 articles from the dataset and examine the agent's performance on correctly including the true author in the candidate pool.

Table 2 reports the percentage of test cases in which the correct author appears in the agent's preliminary list when it is instructed to return 20, 50, or 100 candidates. Overall, this approach to narrowing candidates is ineffective. With 20 candidates, the agent succeeds in only 1.2% of cases. Increasing the list to 50 or 100 improves performance, but makes the downstream analysis exponentially more expensive and therefore impractical.

Database-Augmented Search. When we incorporate the database component of the agent and allow it to warm up by initially collecting a large pool of potential authors, performance improves substantially. For efficiency, we construct the database from a subset of the dataset, disjoint from the evaluation data. We also omit some authors to simulate real-world scenarios where the database does not cover all possible authors. With the database integrated, the search stage becomes much more efficient than starting from scratch in the zero-shot setting. Keyword and embedding similarity search within the database also improves accuracy compared to the more costly web search function. This is reflected in search performance: with 20 candidates, coverage increases to 78.1%, and with 100 candidates, it reaches 93.1%.

Since the database already contains prior articles with precomputed features, computational complexity increases only linearly with the number of candidates. In contrast, in the zero-shot setting, computational cost grows exponentially, as relevant samples must be retrieved and analyzed for each candidate. As a result, the matching process is more efficient and can accommodate a larger number of candidates.

[2] https://huggingface.co/datasets/rjac/all-the-news-2-1-Component-one.

Table 3: Zero-shot attack performance with different matching strategies in the targeted attack scenario with different numbers of samples per candidate and the number of targets. ES = Embeddings Similarity, LDA = LLM Direct Analysis, SALA = Stylometry-Assisted LLM Analysis, No.= Number of.

No. Target	No. Sample	ES	LDA	SALA
1	5	0.500	0.500	0.619
	10	0.513	0.500	0.645
3	5	0.167	0.168	0.244
	10	0.172	0.166	0.291

Table 4: Zero-shot attack performance in an open-world setting.

	ES	LDA	SALA
Full	0.000	0.000	0.000
Filtered	0.053	0.058	0.068

4.4 Zero-Shot Match Results

We first analyze the agent's zero-shot candidate-matching performance without using the database module across the two attack scenarios.

Targeted Attack. Table 3 reports the F1 score for predicting whether a given article was written by the target author. We evaluate three methods while varying the number of target authors and the number of writing samples per target. Directly comparing embedding similarity is ineffective. This is expected, as subtle stylistic differences are unlikely to be captured by embeddings, which are better suited to topical comparisons. The LDA approach is also ineffective. By analyzing the responses, we find that when conducting pairwise comparisons, the LLM hallucinates and consistently determines that the two articles differ significantly in style, even when they are both news articles written in a very formal style. Relying solely on the LLM to analyze texts and generate descriptions is therefore unreliable. SALA achieves the best performance in this setting. The stylometric statistics used in SALA provide grounding, helping the LLM produce more reliable analyses. However, overall performance remains limited, with a maximum F1 score of 0.645. This is likely due to the limited number of available samples for matching, constrained by the high computational cost of the zero-shot setting.

Open-World Attack. As seen from Table 2, without the memory module, the potential candidate selection already has very limited success. Making further analysis based on this limited success in candidate selection does not yield any successful predictions, regardless of the matching strategy. Table 4 shows that the full pipeline fails to produce any correct predictions. To isolate the matching stage, we filter out cases where the correct author is not among the selected candidates and evaluate performance in the zero-shot setting. Given the correct candidates, the agent can make correct predictions, but performance remains low across all matching strategies. More detailed analysis requires the database module to first improve overall prediction performance.

Table 5: Authorship inference performance with different matching strategies in targeted attack scenario with different numbers of samples per candidate and the number of targets.

No. Target	No. Sample	ES	LDA	SALA
1	10	0.500	0.654	0.674
	200	0.500	0.699	0.801
	500	0.500	0.681	0.827
3	10	0.167	0.243	0.312
	200	0.167	0.223	0.551
	500	0.167	0.198	0.595

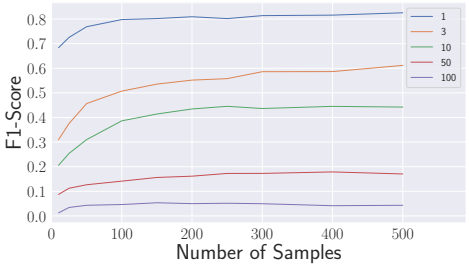

Figure 1: Authorship inference performance in targeted attack scenario with different numbers of samples per candidate and number of targets.

4.5 Database-Augmented Match Results

The incorporation of a database into the agent not only improves candidate search efficacy and efficiency, as shown in subsection 4.3, but also improves downstream matching performance by allowing a significantly larger number of samples to be used for analysis and matching. We now present the agent's authorship inference performance in the two scenarios with the database module implemented.

Targeted Attack. In the *targeted attack* scenario, introducing a database substantially increases the number of prior article samples per candidate, up to 500. This is feasible because the web search

Table 6: Authorship inference performance when using only one type of stylometric feature in SALA.

Number of Targets	1	3	10
Lexical	0.728	0.524	0.298
Syntactic	0.62	0.446	0.308
Readability	0.545	0.199	0.120
Semantic	0.579	0.245	0.089
Style	0.654	0.188	0.094
Full	0.827	0.595	0.363

Table 7: Leave-one-out ablation for stylometric feature groups in SALA, when the specified feature category is removed while all other features are retained.

Number of Targets	1	3	10
Lexical	0.742	0.517	0.294
Syntactic	0.756	0.533	0.307
Style	0.764	0.541	0.315
Readability	0.809	0.584	0.353
Semantic	0.801	0.577	0.347
Full	0.827	0.595	0.363

Table 8: Percentage of correct inference with database module augmented in the open-world attack scenario.

	Top-1		Top-3	
	Full	**Filtered**	**Full**	**Filtered**
ES	0.035	0.051	0.077	0.112
LDA	0.042	0.056	0.092	0.123
SALA	0.078	0.125	0.156	0.224

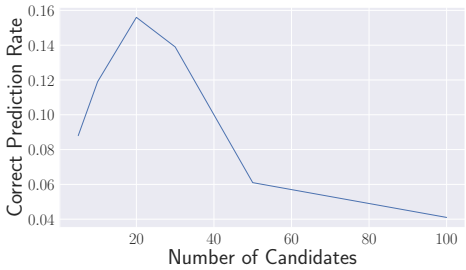

Figure 2: Correct prediction rate of authorship inference (top-3) with database module augmented in the open-world scenario with different numbers of candidates selected during the search stage.

step is bypassed, and all samples are pre-analyzed and aggregated. As a result, the system has a more comprehensive representation of each author while reducing the number of pairwise comparisons.

Table 5 reports the F1 scores for predicting whether a given article is written by the target author. Increasing the sample size significantly improves performance for both LDA and SALA, as both benefit from richer author profiles. With more samples analyzed individually, LDA also becomes more stable and exhibits fewer hallucinations. Our SALA method benefits substantially from the expanded dataset, achieving an F1 score of 0.827. The larger set of references allows stylometric features to better capture each author's writing style.

The correlation between performance and the number of samples per candidate is further illustrated in Figure 1. Performance improves as more samples are added, but plateaus around 300 samples across different numbers of target authors. This suggests that while additional data helps, a moderate number of samples is sufficient for near-optimal performance.

Feature Analysis. With sufficiently strong authorship inference performance, we next analyze finer-grained feature contributions. For the SALA method, we investigate which feature categories are most influential for authorship prediction. Table 6 presents the targeted attack results when SALA utilizes only one feature type at a time. While several feature categories provide moderate predictive power individually, none alone achieves the performance of the full model.

To further examine feature importance and possible redundancy, we additionally conduct a leave-one-out ablation study, shown in Table 7. In this setting, SALA uses all feature categories except the one removed. Removing lexical features results in the greatest performance degradation, followed by syntactic and style features. In contrast, removing readability or semantic features produces only minor changes in performance.

Taken together, the results from both analyses indicate that lexical, syntactic, and stylistic signals contribute most strongly to authorship attribution in our setting, while readability and semantic features provide complementary but less critical information. This behavior is expected given that our dataset consists primarily of news articles, where readability and sentiment-related characteristics tend to be relatively homogeneous across authors.

Open-World Attack. For the more challenging *open-world* attack scenario, Table 8 shows that even with the complete automated pipeline (i.e., search followed by matching), the agent augmented with the database can still make correct predictions, particularly when using our proposed SALA matching strategy. In addition to the increased efficacy of both SALA and LDA with larger numbers of available samples (as demonstrated previously),

the substantial improvement in search performance ensures that enhanced matching accuracy is not offset by incorrect initial candidate selection. Although the overall prediction performance remains relatively low, this is expected given the inherent difficulty of the open-world setting.

Unlike the targeted attack scenario, the number of candidate authors also affects the attack performance. As the number of candidates increases, matching becomes more challenging since selecting the correct author from a larger pool is less likely. Conversely, overly restricting the candidate set may degrade performance if the true author is omitted during the search stage. As illustrated in Figure 2, inference performance peaks when the number of candidates is set at a balanced level, large enough to include the correct author, yet small enough to maintain discriminative power.

Overall, the improved inference accuracy observed across both targeted and open-world settings highlights the significant deanonymization risks, particularly when augmented with a database module that aggregates and pre-processes extensive author profiles.

5 Anonymization Enhancement

Given the strong authorship inference capabilities demonstrated in the previous section, it is important to develop effective countermeasures to mitigate deanonymization risks. While many commercial LLMs include safety guardrails, preventing deanonymization is inherently difficult due to the ambiguity of the task.

Limitations of Guardrails. In practice, we observe that only explicit requests for deanonymization (e.g., "Deanonymize the author of this article") are reliably blocked. However, slightly rephrased requests (e.g., "Compare this article's style to known authors") are often permitted. This highlights a fundamental limitation: malicious intent can be easily obfuscated through minor prompt modifications, even without advanced jailbreak techniques (Li et al., 2023; Deng et al., 2023; Yu et al., 2023; Chao et al., 2023; Liu et al., 2023b; Huang et al., 2023; Wei et al., 2023).

Mitigation Strategy. Our agent provides not only attribution capabilities but also insight into the features that enable deanonymization. We leverage the *Result Reflection Stage* to identify the most revealing stylistic signals and use this information to guide targeted rewriting through the *Anonymization Enhancement Module*.

Table 9: Authorship inference effectiveness with mitigation and content preservation. Average ROUGE-L is reported between the original and rewritten articles.

	Targeted	Open-World	ROUGE-L
No Defense	0.827	0.156	1.000
Direct Paraphrase	0.762	0.035	0.910
Guided Recompose	0.561	0.012	0.899

Results. To assess the effectiveness of our mitigation method introduced in subsection 3.6, we evaluate it under both attack scenarios. We further compare our guided recomposition approach with baseline methods such as direct paraphrasing, in which an LLM is instructed to paraphrase the article without explicit guidance. To ensure that rewriting preserves content, we additionally report ROUGE-L scores between the original and rewritten articles. Table 9 shows that our method substantially reduces authorship inference performance in both settings. In the targeted scenario, F1 decreases from 0.827 to 0.561, while in the open-world scenario, accuracy drops to near-random levels (0.012). Compared to direct paraphrasing, guided recomposition is consistently more effective at removing author-identifying signals. At the same time, high ROUGE-L scores indicate that the rewritten articles retain most of their original content.

Overall, by leveraging the *Result Reflection Stage* to inform the *guided recomposition*, our agent offers a practical and interpretable pathway for mitigating deanonymization risks posed by LLM-agent-powered inference systems.

6 Conclusion

We introduce a deanonymization agent for analyzing authorship risks in news articles, based on a four-stage pipeline. Our SALA (Stylometry-Assisted LLM Analysis) method combines stylometric features with LLM reasoning to enable interpretable and accurate authorship inference, outperforming baselines such as LDA and embedding-based similarity measures. Database augmentation further improves both efficiency and performance, particularly in targeted and open-world scenarios. Finally, through guided recomposition, the agent provides actionable anonymization strategies that mitigate deanonymization risk while preserving content utility.

Limitations

While our proposed framework demonstrates strong performance and interpretability, several limitations remain. First, the current evaluation is restricted to English-language news datasets, which may limit generalizability across domains, genres, or languages. Future work should explore more diverse and variable datasets, including social media, academic writing, and multilingual corpora, to better assess the robustness of the approach. Additionally, although the guided recomposition strategy effectively reduces identifiability, its rewriting quality and stylistic preservation could be further improved through adaptive prompting or human-in-the-loop refinement. Finally, while the database module enhances inference performance, it also introduces potential bias from uneven author representation, suggesting the need for more balanced or dynamically updated author profiles.

Ethics Statement

This work aims to raise awareness of potential deanonymization risks posed by LLM-based authorship analysis and to promote the development of responsible mitigation strategies. By presenting this framework, we hope to facilitate future research on mitigating potential risks of LLM agents in real-world applications.

Our implementation relies on the GPT-4.1 API primarily for rapid prototyping and reproducibility during the research phase. The datasets used in our experiments consist exclusively of publicly available news articles, and no sensitive or private documents are processed. For real-world deployment scenarios involving sensitive data, we strongly recommend the use of locally hosted or open-source language models to ensure full control over data processing and privacy guarantees.

References

GPT-4.1. https://openai.com/index/gpt-4-1/.

Leonard Bauersfeld, Angel Romero, Manasi Muglikar, and Davide Scaramuzza. 2023. Cracking double-blind review: Authorship attribution with deep learning. *PLOS ONE*, 18(6):e0287611.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2023. Jailbreaking Black Box Large Language Models in Twenty Queries. *CoRR abs/2310.08419*.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. 2024. AgentDojo: A Dynamic Environment to Evaluate Attacks and Defenses for LLM Agents. *CoRR abs/2406.13352*.

Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. 2023. Jailbreaker: Automated Jailbreak Across Multiple Large Language Model Chatbots. *CoRR abs/2307.08715*.

Xiangming Gu, Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Ye Wang, Jing Jiang, and Min Lin. 2024. Agent Smith: A Single Image Can Jailbreak One Million Multimodal LLM Agents Exponentially Fast. In *International Conference on Machine Learning (ICML)*. PMLR.

Xie He, Arash Habibi Lashkari, Nikhill Vombatkere, and Dilli Prasad Sharma. 2024. Authorship attribution methods, challenges, and future research directions: A comprehensive survey. *Information*, 15(3).

Baixiang Huang, Canyu Chen, and Kai Shu. 2024. Can large language models identify authorship? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 445–460.

Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. 2023. Catastrophic Jailbreak of Open-source LLMs via Exploiting Generation. *CoRR abs/2310.06987*.

Zhenlan Ji, Daoyuan Wu, Pingchuan Ma, Zongjie Li, and Shuai Wang. 2024. Testing and Understanding Erroneous Planning in LLM Agents through Synthesized User Inputs. *CoRR abs/2404.17833*.

Tharindu Kumarage, Joshua Garland, Amrita Bhattacharjee, Kirill Trapeznikov, Scott Ruston, and Huan Liu. 2023. Stylometric detection of ai-generated text in twitter timelines. *arXiv preprint arXiv:2303.03697*.

LangChain. https://www.langchain.com/.

Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, and Yangqiu Song. 2023. Multi-step Jailbreaking Privacy Attacks on ChatGPT. *CoRR abs/2304.05197*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023a. AgentBench: Evaluating LLMs as Agents. *CoRR abs/2308.03688*.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2023b. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models. *CoRR abs/2310.04451*.

Marcus Ma, Duong Minh Le, Junmo Kang, Yao Dou, John Cadigan, Dayne Freitag, Alan Ritter, and Wei Xu. 2025. Crossnews: a cross-genre authorship verification and attribution benchmark. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press.

Lorenz Mindner, Tim Schlippe, and Kristina Schaaff. 2023. Classification of human-and ai-generated texts: Investigating features for chatgpt. In *International conference on artificial intelligence in education technology*, pages 152–170. Springer.

Lingbo Mo, Zeyi Liao, Boyuan Zheng, Yu Su, Chaowei Xiao, and Huan Sun. 2024. A Trembling House of Cards? Mapping Adversarial Attacks against Language Agents. *CoRR abs/2402.10196*.

Arvind Narayanan, Hristo S. Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. 2012. On the Feasibility of Internet-Scale Author Identification. In *IEEE Symposium on Security and Privacy (S&P)*, pages 300–314. IEEE.

Chidimma Opara. 2024. Styloai: Distinguishing ai-generated content with stylometric analysis. In *International conference on artificial intelligence in education*, pages 105–114. Springer.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. 2024. Identifying the Risks of LM Agents with an LM-Emulated Sandbox. In *International Conference on Learning Representations (ICLR)*. ICLR.

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. 2023. Jailbreak and Guard Aligned Language Models with Only Few In-Context Demonstrations. *CoRR abs/2310.06387*.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. 2023. The Rise and Potential of Large Language Model Based Agents: A Survey. *CoRR abs/2309.07864*.

Wenkai Yang, Xiaohan Bi, Yankai Lin, Sishuo Chen, Jie Zhou, and Xu Sun. 2024. Watch Out for Your Agents! Investigating Backdoor Threats to LLM-Based Agents. *CoRR abs/2402.11208*.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2023. GPTFUZZER: Red Teaming Large Language Models with Auto-Generated Jailbreak Prompts. *CoRR abs/2309.10253*.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. 2024. InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents. *CoRR abs/2403.02691*.

A Prompt Templates

This appendix lists the prompt templates used in each stage of the authorship attribution agent described in section 3. Variables enclosed in brackets (e.g., `[ARTICLE]`) are replaced with task-specific inputs.

A.1 Information Extraction

Extract structured metadata from the following article.
Return a JSON object with the fields:

- topic
- publication_date
- geographic_focus
- publisher_cues

Article:
`[ARTICLE]`

A.2 Candidate Search

Based on the following article metadata, identify up to `[K]` possible authors.
Metadata:

- Topic: `[TOPIC]`
- Date: `[DATE]`
- Location: `[LOCATION]`
- Publisher cues: `[PUBLISHER]`

For each candidate author provide:

- name
- affiliated outlet
- short justification

A.3 Stylometry-Assisted Matching (SALA)

Compare the target article with the candidate author's writing profile.
Target stylometric features:

- lexical: `[LEXICAL_FEATURES]`
- syntactic: `[SYNTACTIC_FEATURES]`
- readability: `[READABILITY]`
- semantic: `[SEMANTIC]`
- style: `[STYLE]`

Candidate profile and reference samples:
`[AUTHOR_PROFILE]`
`[REFERENCE_TEXTS]`
Return:

- authorship score in $[0, 1]$
- key matching features
- key mismatching features
- short explanation

A.4 Result Reflection

Given the attribution results below, identify which stylistic signals enabled authorship inference.
Matching analysis:
`[MATCHING_RATIONALES]`
Target article:
`[ARTICLE]`
Return:

- risk level (Low / Moderate / High)
- most revealing stylistic signals
- signals weakening attribution
- brief explanation

A.5 Anonymization Enhancement

Rewrite the following article to reduce the stylistic signals listed below while preserving meaning.
Signals to reduce:
`[DISCLOSING_FEATURES]`
Article:
`[ARTICLE]`
Requirements:

- preserve factual content
- maintain readability
- avoid introducing new information
- reduce identifiable stylistic patterns

Return the rewritten article.

B Data Contamination Analysis

Since our framework relies on a foundation model and external retrieval, it is important to assess whether attribution performance is influenced by data contamination, such as memorized content or near-duplicate articles. We conduct additional experiments under controlled settings to evaluate this risk.

B.1 Evaluation Setup

Temporal Split. We enforce a time-based split where candidate reference articles are restricted to those published strictly before the target article, reducing overlap in event coverage.

Sanitized Input. We remove explicit identity cues from the target articles, including bylines and publisher indicators.

No-Retrieval Setting. We evaluate a closed-book variant of the agent where candidate search is disabled, requiring the model to rely only on internal knowledge.

B.2 Results

Across all controlled settings, we observe only minor performance changes compared to the default configuration. The temporal split and sanitization introduce negligible degradation, suggesting that attribution is not driven by near-duplicate content or explicit identity cues. Even in the no-retrieval setting, performance remains comparable, indicating

Table 10: Authorship attribution performance under contamination-controlled settings.

Setting	Targeted (F1-Score)	Open-World (Top-1)	Open-World (Top-3)
Default	0.827	0.125	0.243
Temporal Split	0.821	0.121	0.238
Sanitized Input	0.816	0.118	0.234
No Retrieval	0.798	0.109	0.221

that the model primarily relies on stylistic signals rather than memorized content.

Overall, these results suggest that the observed attribution performance is largely robust to potential data contamination effects. While it is difficult to fully rule out memorization in proprietary foundation models, the minimal performance differences across settings indicate that the model's predictions are not dominated by memorized examples or trivial cues.

C Evaluation Beyond News

Authorship attribution in academic writing is also closely related to anonymity concerns, particularly in settings such as double-blind peer review and anonymous preprints, where preserving author identity is important. To assess the applicability of our framework beyond news articles, we conduct a small-scale evaluation on machine learning (ML) research papers.

C.1 Setup

We collect 1,000 papers from major ML conferences. Given a target paper, we adapt the pipeline as follows:

Candidate Search. Instead of news metadata, we use research-specific signals to narrow candidates, including topic keywords, subfields, and research group or affiliation cues (when available).

Candidate Matching. We extend SALA with domain-relevant evidence, including key text patterns (e.g., terminology usage, writing style in methodology sections) and citation connections between authors and referenced works, in addition to standard stylometric features.

C.2 Results

In the open-world setting, the framework achieves *9.1%* Top-3 accuracy on this dataset. Given the substantially larger and more homogeneous author space in academic writing, this performance is non-trivial and comparable to the relative improvements observed in the news domain. These results suggest that the proposed framework generalizes beyond news articles with minimal modification. While the absolute performance is lower due to the more uniform writing style of academic papers, the agent remains capable of identifying useful authorship signals through domain-adapted retrieval and matching.

Overall, this experiment provides preliminary evidence that the framework is applicable to broader text domains, though further large-scale evaluation is left for future work.