Tracing the Roots: A Multi-Agent Framework for Uncovering Data Lineage in Post-Training LLMs

Yu Li[1,2], Xiaoran Shang[2], Qizhi Pei[2,3], Yun Zhu[2], Xin Gao[2,4], Honglin Lin[2,4], Zhanping Zhong[2,4], Zhuoshi Pan[2], Zheng Liu[2], Xiaoyang Wang[2], Conghui He[2], Dahua Lin[2,5], Feng Zhao[1]*, Lijun Wu[2]*

[1]University of Science and Technology of China [2]Shanghai Artificial Intelligence Laboratory
[3]Renmin University of China [4]Shanghai Jiao Tong University
[5]The Chinese University of Hong Kong

liyu01@mail.ustc.edu.cn, fzhao956@ustc.edu.cn, wulijun@pjlab.org.cn

https://arena.opendatalab.org.cn/data-lineage/website/index.html

Abstract

Post-training data plays a pivotal role in shaping the capabilities of Large Language Models (LLMs), yet datasets are often treated as isolated artifacts, overlooking the systemic connections that underlie their evolution. To disentangle these complex relationships, we introduce the concept of **data lineage** to the LLM ecosystem and propose an automated multi-agent framework to reconstruct the evolutionary graph of dataset development. Through large-scale lineage analysis, we characterize domain-specific structural patterns, such as vertical refinement in Math-oriented datasets and horizontal aggregation in General-domain corpora. Moreover, we uncover pervasive systemic issues, including *structural redundancy* induced by implicit dataset intersections and the *propagation of benchmark contamination* along lineage paths. To demonstrate the practical value of lineage analysis for data construction, we leverage the reconstructed lineage graph to create a *lineage-aware diversity-oriented dataset*. By anchoring instruction sampling at upstream leaf sources, this approach mitigates downstream homogenization and hidden redundancy, yielding a more diverse post-training corpus. We further highlight lineage-centric analysis as an efficient and robust topological alternative to sample-level dataset comparison for large-scale data ecosystems. By grounding data construction in explicit lineage structures, our work advances post-training data curation toward a more systematic and controllable paradigm.

1 Introduction

High-quality post-training data is the primary engine driving LLM capabilities (Zhou et al., 2023; Cai et al., 2025a), yet the community lacks systematic mechanisms to track its provenance. While recent efforts have extensively traced the evolution of model architectures (Sajjadi Mohammadabadi et al., 2025; Zhao et al., 2025), datasets are still predominantly treated as isolated artifacts, obscuring their true developmental context. In practice, modern post-training corpora are rarely constructed from scratch; instead, they emerge through recursive derivation processes that repurpose existing resources via semantic evolution (Xu et al., 2024), knowledge distillation (Mitra et al., 2024), and structured fusion (Pei et al., 2025b). As a result, post-training datasets collectively form a dense and interdependent evolutionary network whose connections are largely undocumented.

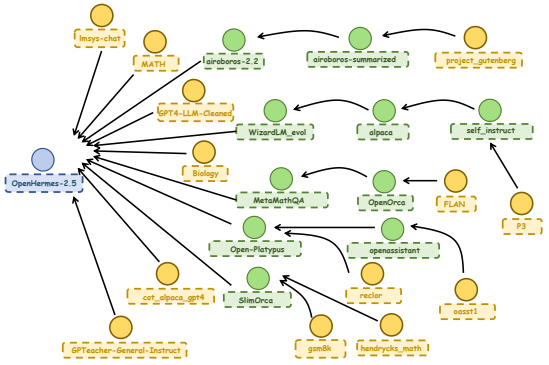

Figure 1: Lineage graph construction with depth 3, using the OpenHermes-2.5 dataset (Teknium, 2023) as an example. Yellow and green nodes denote leaf and internal nodes, respectively.

This lack of lineage transparency raises two risks. First, *structural redundancy* emerges when datasets implicitly inherit overlapping sources, causing downstream corpora to converge semantically despite apparent scale growth (Zhou et al., 2025). Such hidden intersections erode effective diversity and weaken the marginal value of additional data. Second, the *propagation of benchmark contamination* becomes unavoidable when test samples embedded in upstream datasets are unknowingly inherited by downstream derivatives, introducing latent leakage that undermines the credibility of

*Corresponding authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9606–9625

future evaluations (Sainz et al., 2023). Without explicit lineage awareness, these risks remain difficult to detect or mitigate at scale.

To address this challenge, we introduce the concept of **data lineage** and propose a multi-agent collaborative framework that autonomously mines unstructured documentation and performs self-verified provenance tracing. Starting from 83 high-impact seed datasets spanning four major domains, we construct an evolutionary graph comprising 430 unique nodes connected by 971 inheritance edges (a real visualization example is shown in Figure 1). We analyze this ecosystem from three complementary perspectives: *topological structure*, *cross-domain dependencies*, and *temporal evolution*. This multi-view analysis reveals distinct development patterns. General-domain datasets primarily expand horizontally, forming a wide and shallow structure (average depth 1.05) that exhibits signs of saturation. In contrast, Math evolves vertically (average depth 2.92), driven by intensive reuse of core anchors to support deep recursive refinement. Cross-domain analysis further identifies Code as a critical bridge between General and Math, while highlighting the severe scarcity of specialized Science data (only 44 nodes), which necessitates heavy reliance on upstream resources from other domains.

This explicit lineage transparency enables concrete diagnoses of structural issues. Our analysis shows that 17 of the 83 examined datasets exhibit redundancy rates exceeding 1%, with `open-instruct-v1` (hakurei, 2023) reaching 46.48% due to the inclusion of its own superset. Moreover, we uncover widespread benchmark contamination propagation: 19 datasets demonstrate varying degrees of leakage across benchmarks such as `Omni-MATH` (Gao et al., 2025a), `TheoremQA` (Chen et al., 2023), `LiveCodeBench` (Jain et al., 2025), `TruthfulQA` (Lin et al., 2022), and `SciBench` (Wang et al., 2024). A notable example is `Caco-1.3M` (Lin et al., 2025), which implicitly inherits 37.95% of `Omni-MATH` samples from contaminated upstream sources despite not explicitly including the benchmark itself. Unlike conventional sample-based scanning, our lineage-based framework exposes these latent structural intersections and enables efficient tracing of contamination sources along inheritance paths.

Beyond diagnosis, we demonstrate the practical value of data lineage analysis by using it to guide the construction of a *lineage-aware diversity-oriented dataset* via provenance-based sampling, which anchors selection at upstream leaf sources to explicitly counteract redundancy induced by derivative reuse. Furthermore, we include a discussion on how lineage-centric analysis enables a shift from sample-level comparison to topological reasoning over dataset evolution, offering advantages in matching efficiency, robustness to semantic drift, discovery of evolutionary patterns, and long-term ecosystem scalability.

Our contributions are summarized as follows: (1) We introduce the concept of data lineage and propose a multi-agent framework to reconstruct the evolutionary dependencies of post-training datasets. (2) We analyze the ecosystem to characterize domain-specific evolution and reveal structural issues, specifically quantifying data redundancy and tracing the propagation of benchmark contamination. (3) We propose a lineage-aware curation strategy to maximize query-level semantic diversity, achieving superior diversity metrics compared to datasets across varying scales.

2 Related Work

2.1 Post-Training Data Construction

Post-training data acquisition has evolved from the early aggregation of real-world annotations (Longpre et al., 2023; Hendrycks et al., 2021; Cobbe et al., 2021; Raffel et al., 2020; Weber et al., 2024) to a multidimensional synthesis paradigm. Dominant strategies now include *semantic evolution* (Xu et al., 2024; Luo et al., 2025; Pei et al., 2025a; Gao et al., 2025b) for complexity enhancement, *knowledge distillation* (Mitra et al., 2024; Tian et al., 2025; Guha et al., 2026) leveraging teacher CoT traces, and *structured fusion* (Pei et al., 2025b; Pan et al., 2025) for composite reasoning for distribution refinement, alongside multimodal augmentations (Shen et al., 2025; Yu et al., 2025). Consequently, data constructed entirely "from scratch" has become rare (Xu et al., 2025; Li et al., 2025c). This widespread repurposing yields deeply nested dependencies, but their evolutionary pathways are seldom tracked.

2.2 Data Analysis Paradigms and the Evolution of Sourcing

In response to this entangled landscape, analysis tools have evolved from early documentation initiatives (Google, 2021; Luccioni et al., 2021) to

dataset quality evaluation and policy/compliance checks. Current approaches range from quality-based filtering (Liu et al., 2024; Chen et al., 2024; Lu et al., 2024) and cross-domain mixing analysis (Li et al., 2025b) to large-scale corpus profiling and licensing audits (Elazar et al., 2024; Longpre et al., 2024). Although the concept of "sourcing" has been effectively operationalized to trace model architectural history (Zhao et al., 2025) or attribute specific model behaviors to individual training instances (Akyurek et al., 2022; Guu et al., 2023; Pang et al., 2025), these methods typically focus on the internal analysis of isolated datasets or specific samples, leaving the evolutionary relationships between datasets in the data ecosystem largely unexplored.

3 Data Lineage

Preliminary. We introduce a systematic framework for the automated tracing of data lineage, which is formally defined as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node in $\mathcal{V}$ corresponds to a post-training dataset. The nodes are categorized into two types: *internal node*, which is the dataset with identifiable upstream sources that enable recursive tracing; and *leaf node*, which denotes a terminal dataset that lacks such prerequisites and define the boundaries of automated exploration. Directed edges in $\mathcal{E}$ represent inheritance dependencies, where an edge $(v_i, v_j) \in \mathcal{E}$ indicates that the upstream dataset v_i contributes to the construction of the dataset v_j.

3.1 Challenges in Lineage Tracing

Tracing lineage at scale is non-trivial due to the informal and heterogeneous nature of dataset documentation. Provenance information is scattered across various sources, such as academic papers, repository READMEs, and technical blogs, and is rarely expressed in a standardized format. Moreover, the dependency structure is often extensive and deeply nested: a single dataset may cite numerous upstream sources, and recursively expanding these references risks a combinatorial explosion in the search space. To address these challenges, we design a multi-agent collaborative framework that coordinates multi-source evidence fusion and semantic reasoning to extract structured lineage from noisy, incomplete documentation.

3.2 Multi-Agent Collaborative Framework

As illustrated in Figure 2, our framework operates via a target-to-source recursive pipeline managed through a centralized processing queue of pending datasets, designed to incrementally construct the lineage graph by tracing upstream dependencies. For each candidate dataset, the pipeline executes four sequential steps:

(1) Candidate Validation. We initialize the framework by enqueuing all candidate datasets into the centralized processing queue. For each candidate, we first filter out previously processed entries to prevent redundant computation and subsequently verify its availability via the HuggingFace API. To address potential latencies between research publication and repository upload, we determine the dataset's effective release time by cross-referencing its HuggingFace timestamp with the publication date of its associated paper, adopting the earlier of the two as the canonical release date. Finally, to align with the modern LLM era following GPT-3 (Brown et al., 2020), we restrict our analysis to datasets with an effective release time after 2020.

(2) Multi-source Information Retrieval. For each validated candidate dataset, we issue a request to retrieve its HuggingFace README. We employ a *sourcing agent* to parse the documentation and discover external resources, including GitHub repositories, technical blogs, and papers. Subsequently, we dispatch specialized *extracting agents* to fetch the associated content. Specifically, agents retrieve web content for repositories and blogs, while querying the arXiv API for papers using titles or URLs. To enhance context quality and mitigate interference during subsequent lineage analysis, we apply a tailored filtering mechanism to eliminate structural noise such as metadata headers and code blocks in READMEs, HTML tags in blogs, and non-informative sections in papers. Finally, the curated materials are consolidated into a unified resource context.

(3) Semantic Source Inference and Extraction. Building on the consolidated resource context, we deploy a pool of *tracing agents* operating in parallel to identify the source data utilized in constructing the candidate dataset. These agents are explicitly instructed to distinguish actual sources from incidental mentions, rigorously excluding entities such as evaluation benchmarks, comparison baselines, and non-integrated references. The ex-

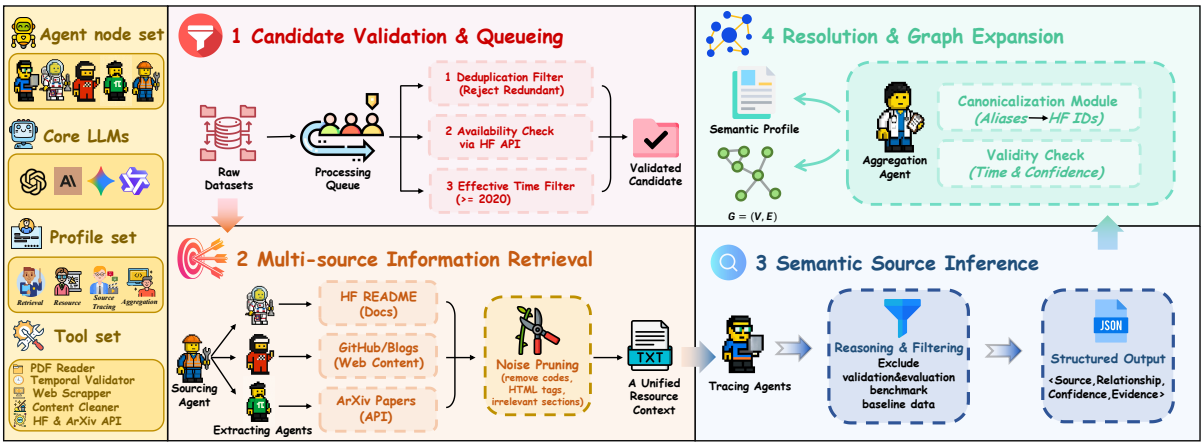

Figure 2: Overview of the multi-agent data lineage reconstruction framework. The system coordinates collaborative agents to extract provenance information from unstructured documentation, transforming isolated datasets into a comprehensive evolutionary graph.

traction results are formalized as structured JSON records $\langle S, R, C, E \rangle$, where S identifies the constituent source ancestor; R categorizes the specific derivation relation (e.g., CoT distillation, question reformulation); C quantifies identification confidence based on textual support strength and source credibility; and E captures the supporting evidence. These records are aggregated to instantiate directed edges in the lineage graph.

(4) Aggregation, Resolution, and Recursive Expansion. The raw extraction records from parallel agents are first pooled and deduplicated by an *aggregation agent* to eliminate redundancy across documentation sources. To address naming inconsistencies, the agent employs a retrieval-augmented resolution module that attempts to canonicalize informal aliases into unique HuggingFace IDs (i.e., org/name) via API verification and similarity reasoning. We subsequently enforce rigorous validity checks, pruning anachronistic edges where the source postdates the target and filtering out low-confidence hallucinations lacking verifiable evidence. Beyond structural lineage, the agent synthesizes a comprehensive semantic profile for the target dataset by integrating its inherent metadata with the composition of its upstream sources. This profile encapsulates key attributes including the dataset summary, capability domains, and construction methods. Finally, identified upstream sources are submitted to the centralized processing queue for subsequent recursive processing.

Graph Construction and Verification Strategy. Anchored within the ecosystem, our framework utilizes canonical org/name identifiers to execute a Depth-First Search (DFS) traversal over the dependency network. This recursion terminates at leaf nodes, identified by two convergence criteria: (1) foundational status lacking ancestors (upstream sources); or (2) release dates predating 2020. While preserved for completeness, these nodes halt further expansion. Crucially, to mitigate LLM hallucinations, we implement a confidence-aware expert verification protocol that automatically routes low-confidence extractions for manual review, ensuring the integrity of the final ecosystem map.

4 Landscape Analysis

Experimental Setup. We implement the lineage tracing framework using LangChain[1] for workflow orchestration, leveraging GPT-5.1[2] and Gemini-2.5-flash (Comanici et al., 2025) as the underlying models for agent implementation. Notably, once deployed, the provenance system can trace newly encountered data in real time. To delineate the current landscape of the post-training ecosystem, we focus our analysis on the textual modality, as it represents the most prevalent data form. For dataset selection, we jointly consider HuggingFace downloads, repository likes, and citations, curating **83** high-impact textual datasets spanning four domains (General, Math, Code, and Science) as seed roots for recursive traversal. Dataset details and framework configurations are provided in Appendix A.

[1] https://github.com/langchain-ai/langchain

[2] specifically version `gpt-5.1-2025-11-13`; see https://platform.openai.com/docs/models/gpt-5.1.

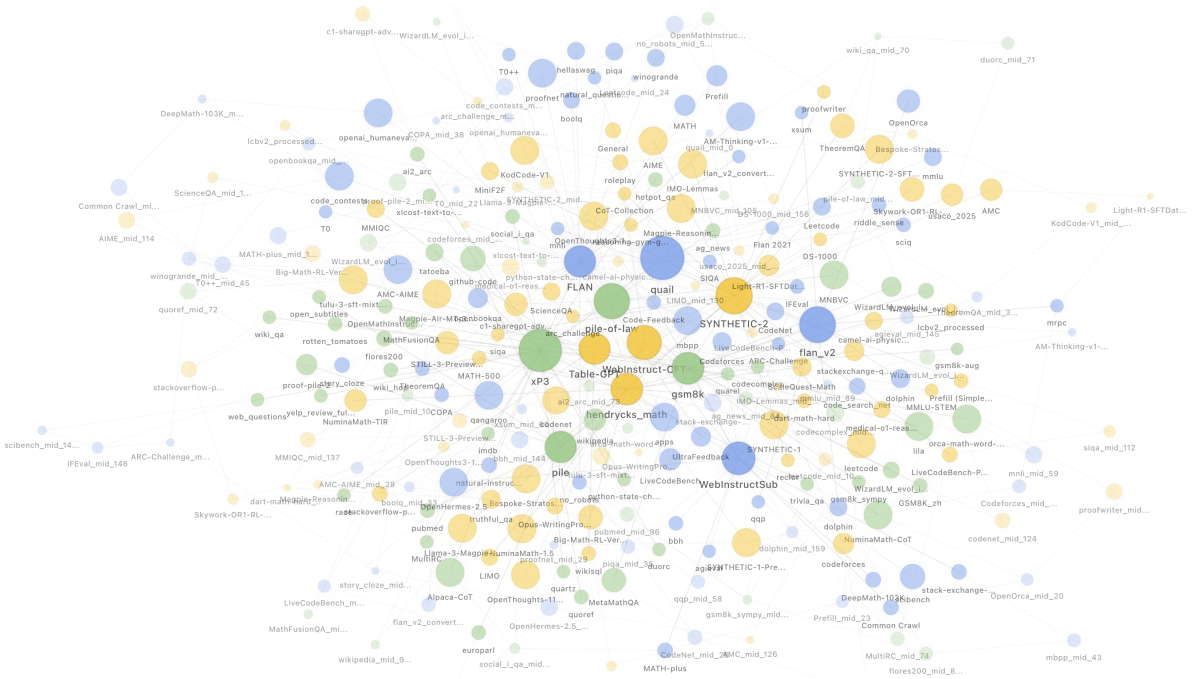

Figure 3: Partial high-level overview of data lineage relationships, where node size reflects data download count, colors represent distinct data sub-networks, and darker shades indicate higher-degree, more important nodes.

Domain	Nodes	Depth	In-Deg.	Out-Deg.	Leaf %
Math	99	**2.92**	3.30	**1.54**	38.38%
Code	98	2.12	3.78	1.36	43.88%
General	**285**	1.05	2.51	1.29	**68.42%**
Science	44	2.82	**3.98**	1.25	47.73%

Table 1: Topological statistics by domain.

Source→Target	Math	Code	General	Science
Math	147 (44.82%)	63 (17.80%)	48 (9.66%)	43 (23.24%)
Code	67 (20.43%)	118 (33.33%)	65 (13.08%)	48 (25.95%)
General	73 (22.26%)	137 (38.70%)	350 (70.42%)	64 (34.59%)
Science	41 (12.50%)	36 (10.17%)	34 (6.84%)	30 (16.22%)

Table 2: Cross-domain dependency matrix reordered by domain. Values denote counts (and column-wise percentages), indicating the proportion of a target domain's composition derived from each source domain.

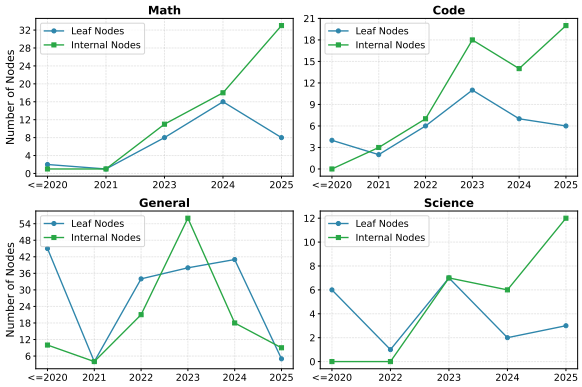

Figure 4: Temporal distribution of data lineages by domain. The plots illustrate the number of datasets released each year, categorized by node type.

4.1 Global Topology and Evolutionary Trends

Starting from 83 high-impact seed datasets, our recursive lineage tracing reconstructed an expansive graph comprising 430 distinct datasets connected by 971 inheritance edges, with partial data lineage relationships illustrated in Figure 3. By synthesizing graph topological metrics, cross-domain composition dependencies, and temporal evolutionary trajectories, we identify a fundamental structural transition characterized by divergent evolutionary strategies across domains.

Evolutionary Divergence: Broad Accumulation vs. Deep Refinement. The topology of the lineage graph reveals two differentiated evolutionary patterns across domains. (1) General-domain datasets exhibit a broad accumulation paradigm. Their shallow structures (leaf ratio 68.42% and average depth 1.05, Table 1) reflect a strategy that prioritizes covering a wide range of topics by gathering information from many sources. This tendency to aggregate data is exemplified by massive collections like `FineWeb` ($d_{in} = 111$, Table 7), while even the most reused dataset `FLAN` shows modest downstream influence ($d_{out} = 7$, Table 6). (2) Conversely, Math follows a deep re-

finement paradigm characterized by a vertically structured, recursive topology (average depth 2.92, average $d_{out} = 1.54$). High-consensus anchors like `hendrycks_math` ($d_{out} = 19$) and `gsm8k` ($d_{out} = 14$) serve as central roots supporting multiple generations of descendants. This is clearly observed in `NuminaMath` ($d_{out} = 13$), which evolved from these foundational datasets and has now become a new cornerstone itself. Given this recursive structure, introducing novel, independent mathematical seeds is essential to complement synthetic augmentation and prevent saturation.

Cross-Domain Composition and Functional Specialization. The dependency matrix (Table 2) reveals distinct functional roles within the ecosystem. (1) Self-sourcing and independence. The General domain exhibits the highest independence, evidenced by a dominant 70.42% self-sourcing rate. Math ranks second (44.82%), demonstrating strong internal recycling while drawing on General inputs for linguistic context. (2) Code as the operational link. Code operationalizes reasoning through a balanced profile: 38.70% from General (capturing user intent) and 17.80% from Math (enhancing reasoning capabilities). This dual-sourcing positions Code datasets as a functional intermediary, translating abstract logic into verifiable, executable outputs. (3) Science: Underdeveloped Status and High Dependence. With only 44 nodes, the Science domain remains underdeveloped. Its high external usage (average $d_{in} = 3.98$) paired with low self-sourcing (16.22%) indicates a heavy reliance on other domain resources due to the scarcity of native data. Consequently, future efforts should prioritize constructing specialized foundational datasets to bridge this gap.

Temporal Evolution: General Saturation vs. Specialized Growth. The temporal trajectories (Figure 4) reveal a decisive shift in community prioritization. (1) Saturation of broad acquisition. The General domain exhibits clear signs of saturation. New leaf node injection dropped sharply from 41 in 2024 to just 5 in 2025, marking the relative maturity of foundational natural language coverage and indicating that the phase of broad raw text acquisition has plateaued. (2) Strategic prioritization of specialized reasoning. Conversely, focus has shifted toward specialized domains. Math witnessed a surge in intermediate nodes ($18 \rightarrow 33$ from 2024 to 2025), cementing its status as the core driver for logic enhancement. Similarly, Science saw its intermediate output double ($6 \rightarrow 12$), signaling escalating attention to complex domain challenges. This divergence confirms a structural transition from broad knowledge accumulation to deep reasoning synthesis.

Dataset Name	Rate(%)
open-instruct-v1 (hakurei, 2023)	46.48
opc-sft-stage2 (Huang et al., 2025)	27.96
codeforces-cots (Penedo et al., 2025)	23.12
Python-Code-23k-ShareGPT (Bawase, 2023)	19.89
CodeFeedback-Filtered-Instruction (Zheng et al., 2024)	8.00
OpenMathInstruct-2 (Toshniwal et al., 2025)	6.11
Fast-Math-R1-SFT (Yoshihara et al., 2025)	5.30
Open-Omega-Forge-1M (prithivMLmods, 2025)	4.33
Light-R1-SFTData (Wen et al., 2025)	4.29
OpenCodeReasoning (Ahmad et al., 2025)	3.92

Table 3: Analysis of source intersections across the top-10 datasets, ranked in descending order. Further details are provided in Appendix C.

4.2 Analysis of Source Intersection

As the community shifts toward a data-centric paradigm, dataset iteration has accelerated through the recursive integration of existing high-quality corpora. While this strategy allows developers to "stand on the shoulders of giants"—minimizing cold-start costs while expanding scale—it introduces latent structural risks. Specifically, the lack of lineage transparency leads to unintended structural intersections, where seemingly distinct datasets unknowingly converge on identical upstream sources.

To expose latent structural intersections, our lineage tool constructs a dependency graph revealing upstream connections hidden within deep pathways. Using `Fast-Math-R1-SFT` (Yoshihara et al., 2025) as a case study, our tool detected the concurrent incorporation of `OpenR1-Math-220k` and its superset, `Light-R1-SFTData` (Wen et al., 2025). A similar pattern recurs in `open-instruct-v1`, where `self_instruct` is repeatedly included in its lineage (Figure 5). To quantify this redundancy, we applied a strict metric based on exact (instruction, input, output) triplet matches. This analysis revealed unintended redundancy rates of 5.30% and 46.48%, respectively. Extending this scrutiny to the broader ecosystem, we identified similar nested patterns across multiple collections; the top 10 datasets exhibiting the highest structural redundancy are summarized in Table 3.

In light of these findings, several recommendations are summarized (Rec.): Rec.❶ Prioritize or-

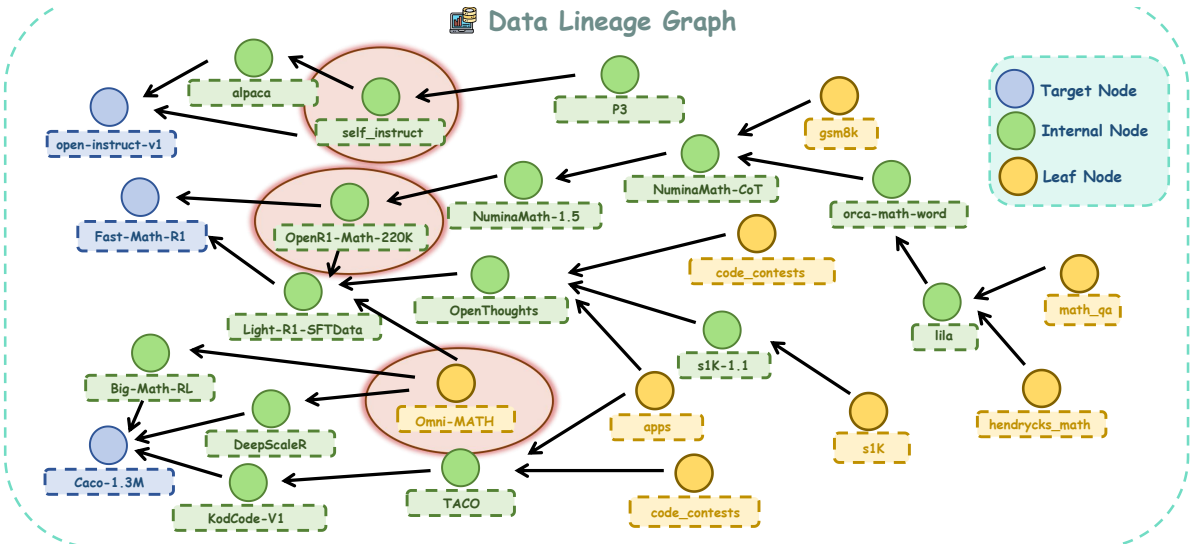

Figure 5: Schematic illustration of a small subgraph of the data lineage graph, showing the partial composition of three target datasets; downstream overlap nodes and benchmark contamination locations are highlighted in **red**.

thogonal datasets via lineage analysis. For example, avoid adding OpenR1 if its superset Light-R1 is already selected, thereby maximizing diversity efficiency. Rec.❷ When multi-path ingestion is unavoidable, immediate deduplication is essential to eliminate structural redundancy caused by intersecting sources.

4.3 Analysis of Benchmark Contamination

Data contamination, defined as the inadvertent inclusion of evaluation data into training corpora, undermines evaluation credibility. This issue creates a cascading effect where pollution in upstream datasets propagates downstream through the construction pipeline. Such propagation blurs the boundary between training and testing data, which creates a false sense of model capability.

Traditional decontamination methods, such as N-gram matching or semantic embedding retrieval (Golchin and Surdeanu, 2024; Li et al., 2024), encounter significant limitations. Although precise, they require computationally expensive sample-wise scans and fail to map propagation across datasets. In contrast, data lineage offers a new, global perspective. It traces contamination diffusion along inheritance paths, which allows users to pinpoint upstream sources without full-scale content scanning. Leveraging our lineage graph, we detect contamination across five benchmarks involving 19 datasets. As shown in Figures 5 and 6, `DeepScaleR-Preview-Dataset` and `Big-Math-RL-Verified` directly ingest `Omni-MATH`, which results in leakage rates

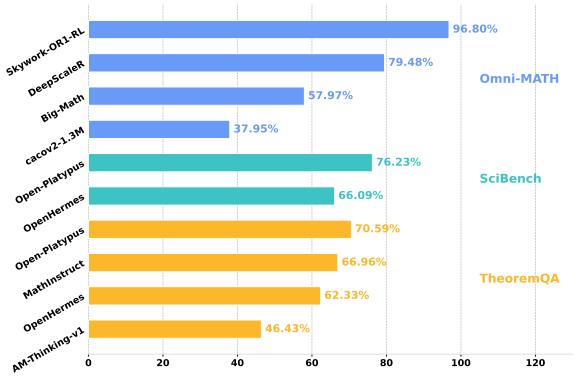

Figure 6: Benchmark contamination analysis across various datasets. Additional contamination results are provided in Appendix D.

of 79.48% and 57.97%, respectively. Consequently, `Caco-1.3M` inadvertently inherits 37.95% contamination by incorporating these datasets.

We propose the following protocols for data curators (Rec.): Rec.❶ Exclude compromised benchmarks. If upstream sources contain benchmark data, strict exclusion of the corresponding evaluation sets is imperative to prevent performance inflation. Rec.❷ Pre-screen via lineage analysis. Before integrating external data, lineage tools audit upstream composition to intercept unintentional leakage at the source.

5 Analysis and Discussion

The global data lineage graph supports both diagnosing issues such as contamination and guiding data curation. We discuss how topological signals

can improve dataset construction, and summarize the benefits of lineage-centric analysis over sample-level methods.

5.1 Lineage-Guided Data Construction

In post-training data construction, the instruction (query) serves as a stable semantic anchor. While response distillation is inherently constrained by teacher models, instructions often remain invariant across multiple rounds of lineage reuse. Consequently, recursive dataset derivation frequently induces implicit redundancy, narrowing the effective problem space (Sandholm et al., 2024).

To address this, we propose *provenance-based sampling*, a lineage-aware strategy designed to maximize instruction diversity. Rather than sampling from the entire corpus, we treat leaf node datasets ($d_{in} = 0$) as upstream knowledge anchors. We prioritize these sources based on both their domain metadata and topological influence, quantified by out-degree. Following this selection, we apply MinHash (Broder et al., 1997) for duplicate removal. This yields a lineage-aware dataset with 570K unique instructions.

Empirical Validation and Potential. We evaluate diversity using the Vendi Score (Friedman and Dieng, 2022) and Centroid Distance (Suwanda et al., 2020). To ensure a rigorous evaluation, we benchmarked our approach against a diverse array of datasets ranging from 300K to 1.2M samples. We selected baselines that are widely recognized for their quality and coverage rather than domain-specific niches. As presented in Table 4, our strategy achieves a Vendi Score of **452.44** and a Centroid Distance of **0.6385**. These results demonstrate that our method delivers superior performance across the entire spectrum of data scales. Notably, our dataset outperforms `OpenHermes-2.5`, a strong baseline renowned in the community for its extensive topic coverage. Even more significantly, our approach substantially exceeds much larger collections such as `MegaScience` and `OpenThoughts3`. Despite these datasets containing more than double the number of samples compared to ours, they exhibit lower diversity scores.

This empirical evidence shows that larger data volume does not automatically translate to higher semantic diversity scores. Our lineage-guided approach achieves superior diversity metrics at a smaller scale, demonstrating the effectiveness of provenance-based sampling in maximizing instruction coverage.

Dataset	Size	Diversity Metric	
		Vendi Score ↑	Cent. Dist. ↑
OmniThought-0528 (Cai et al., 2025b)	301K	162.52	0.5140
herculesv1 (Locutusque, 2024)	463K	397.33	0.6121
OpenHermes-2.5 (Teknium, 2023)	615K	437.76	0.6271
TextbookReasoning (Fan et al., 2025)	651K	283.75	0.5598
MiroMind-M1-SFT (Li et al., 2025a)	719K	108.89	0.4597
tulu-3-sft-mixture (Lambert et al., 2025)	939K	375.78	0.6169
OpenThoughts3 (Guha et al., 2026)	1.2M	133.26	0.4970
MegaScience (Fan et al., 2025)	1.2M	373.78	0.6150
Ours (Provenance-based)	**570K**	**452.44**	**0.6385**

Table 4: Diversity comparison using Vendi Score and Centroid Distance.

Notably, these results are obtained using only leaf nodes: we intentionally exclude internal evolutionary variants to isolate the effect of provenance disentanglement. That this leaf-only subset outperforms complex mixtures suggests two implications.

(1) Efficiency: Preserving original provenance can yield high diversity without exhaustive filtering over the derivative space.

(2) High Ceiling: Since refined internal nodes with richer rewrites and semantic variants are not yet included, there remains substantial headroom; incorporating these hubs may further improve dataset quality.

5.2 Lineage vs. Sample-Level Analysis

Beyond dataset construction, the lineage graph provides a more efficient framework for data analysis compared to traditional sample-level methods. We highlight three key advantages.

Efficiency in Dataset Matching. Sample-level similarity checks require scanning and comparing large numbers of examples, which is costly at scale. Lineage analysis shifts the unit of comparison from samples to ancestry. By comparing the overlap of upstream sources and inheritance paths, we can quickly estimate whether two datasets are likely to share content and where the overlap comes from, without traversing millions of samples.

Discovery of Evolutionary Paradigms. The graph groups related datasets into families and makes their construction steps explicit. By tracing recurring parent-to-child transformations in successful lineages, we can summarize common build patterns, such as textbook sources, synthetic Q&A generation, and CoT refinement. These patterns offer practical guidance on which sources and refinement steps often co-occur in high-impact datasets.

Robustness Against Semantic Drift. Content-based matching is fragile when data is rewritten, expanded, or reformatted, since surface similarity can disappear. Lineage analysis relies on dependency links and provenance records rather than text overlap, so it can still connect an evolved dataset to its origins even after substantial edits. This preserves traceability and supports reliable auditing of redundancy or contamination.

6 Conclusion

In this paper, we introduce a multi-agent collaborative framework to reconstruct a large-scale data lineage graph for post-training datasets. Our analysis uncovers structural redundancy and traces the propagation of benchmark contamination across the ecosystem. Furthermore, we discussed the application of the lineage graph in data construction, focusing on diversity, and outlined potential directions for future research.

Limitations

Our framework faces two primary limitations. First, relying on LLMs entails inherent hallucination risks, necessitating human verification for low-confidence extractions to ensure graph reliability. Second, our lineage reconstruction is strictly bound by documentary transparency; the system cannot recover dependencies if dataset creators fail to report or intentionally conceal upstream sources in their technical documentation.

Acknowledgements

This work was supported by the Anhui Provincial Natural Science Foundation under Grant 2108085UD12, and Shanghai Artificial Intelligence Laboratory. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC. The AI-driven experiments, simulations and model training were performed on the robotic AI-Scientist platform of Chinese Academy of Sciences.

References

Wasi Uddin Ahmad, Sean Narenthiran, Somshubra Majumdar, Aleksander Ficek, Siddhartha Jain, Jocelyn Huang, Vahid Noroozi, and Boris Ginsburg. 2025. OpenCodeReasoning: Advancing data distillation for competitive coding. In *the 2nd Conference on Language Modeling*, pages 1–15.

Ekin Akyurek, Tolga Bolukbasi, Frederick Liu, Binbin Xiong, Ian Tenney, Jacob Andreas, and Kelvin Guu. 2022. Towards tracing knowledge in language models back to the training data. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 2429–2446.

Ajinkya Bawase. 2023. Python-code-23k-sharegpt. https://huggingface.co/datasets/ajibawa-2023/Python-Code-23k-ShareGPT. Dataset.

Andrei Z Broder, Steven C Glassman, Mark S Manasse, and Geoffrey Zweig. 1997. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29(8-13):1157–1166.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Mengzhang Cai, Xin Gao, Yu Li, Honglin Lin, Zheng Liu, Zhuoshi Pan, Qizhi Pei, Xiaoran Shang, Mengyuan Sun, Zinan Tang, Xiaoyang Wang, Zhanping Zhong, Yun Zhu, Dahua Lin, Conghui He, and Lijun Wu. 2025a. OpenDataArena: A fair and open arena for benchmarking post-training dataset value. *Preprint*, arXiv:2512.14051.

Wenrui Cai, Chengyu Wang, Junbing Yan, Jun Huang, and Xiangzhong Fang. 2025b. Reasoning with OmniThought: A large CoT dataset with verbosity and cognitive difficulty annotations. *Preprint*, arXiv:2505.10937.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. AlpaGasus: Training a better alpaca with fewer data. In *the 12th International Conference on Learning Representations*, pages 1–31.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. 2023. TheoremQA: A theorem-driven question answering dataset. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 7889–7901.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Yanai Elazar, Akshita Bhagia, Ian Helgi Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Evan Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hannaneh Hajishirzi, Noah A. Smith, and Jesse Dodge. 2024. What's in my big data? In *the 12th International Conference on Learning Representations*, pages 1–56.

Run-Ze Fan, Zengzhi Wang, and Pengfei Liu. 2025. MegaScience: Pushing the frontiers of post-training datasets for science reasoning. *arXiv preprint arXiv:2507.16812*.

Dan Friedman and Adji Bousso Dieng. 2022. The vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, pages 1–32.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2025a. Omni-MATH: A universal olympiad level mathematic benchmark for large language models. In *the 13th International Conference on Learning Representations*, pages 1–30.

Xin Gao, Qizhi Pei, Zinan Tang, Yu Li, Honglin Lin, Jiang Wu, Lijun Wu, and Conghui He. 2025b. A strategic coordination framework of small LMs matches large LMs in data synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11552–11570.

Shahriar Golchin and Mihai Surdeanu. 2024. Time travel in LLMs: Tracing data contamination in large language models. In *the 12th International Conference on Learning Representations*, pages 1–22.

Google. 2021. Know Your Data.

Etash Kumar Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Rea Sprague, Ashima Suvarna, Benjamin Feuer, Leon Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, and 32 others. 2026. OpenThoughts: Data recipes for reasoning models. In *the 14th International Conference on Learning Representations*, pages 1–72.

Kelvin Guu, Albert Webson, Ellie Pavlick, Lucas Dixon, Ian Tenney, and Tolga Bolukbasi. 2023. Simfluence: Modeling the influence of individual training examples by simulating training runs. *Preprint*, arXiv:2303.08114.

hakurei. 2023. Open instruct v1: A dataset for having LLMs follow instructions. Hugging Face Datasets. Accessed: 2026-01-03.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. *Advances in Neural Information Processing Systems*, pages 1–22.

Siming Huang, Tianhao Cheng, Jason Klein Liu, Weidi Xu, Jiaran Hao, Liuyihan Song, Yang Xu, Jian Yang, Jiaheng Liu, Chenchen Zhang, and 1 others. 2025. OpenCoder: The open cookbook for top-tier code large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33167–33193.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *the 13th International Conference on Learning Representations*, pages 1–41.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, and 4 others. 2025. Tulu 3: Pushing frontiers in open language model post-training. In *the 2nd Conference on Language Modeling*, pages 1–76.

Xingxuan Li, Yao Xiao, Dianwen Ng, Hai Ye, Yue Deng, Xiang Lin, Bin Wang, Zhanfeng Mo, Chong Zhang, Yueyi Zhang, Zonglin Yang, Ruilin Li, Lei Lei, Shihao Xu, Han Zhao, Weiling Chen, Feng Ji, and Lidong Bing. 2025a. MiroMind-M1: An open-source advancement in mathematical reasoning via context-aware multi-stage policy optimization. *Preprint*, arXiv:2507.14683.

Yu Li, Zhuoshi Pan, Honglin Lin, Mengyuan Sun, Conghui He, and Lijun Wu. 2025b. Can one domain help others? a data-centric study on multi-domain reasoning via reinforcement learning. *Preprint*, arXiv:2507.17512.

Yu Li, Qizhi Pei, Mengyuan Sun, Honglin Lin, Chenlin Ming, Xin Gao, Jiang Wu, Conghui He, and Lijun Wu. 2025c. CipherBank: Exploring the boundary of LLM reasoning capabilities through cryptography challenge. In *Findings of the Association for Computational Linguistics: ACL*, pages 5929–5965.

Yucheng Li, Yunhao Guo, Frank Guerin, and Chenghua Lin. 2024. An open-source data contamination report for large language models. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 528–541.

Honglin Lin, Qizhi Pei, Zhuoshi Pan, Yu Li, Xin Gao, Juntao Li, Conghui He, and Lijun Wu. 2025. Scaling code-assisted chain-of-thoughts and instructions for model reasoning. In *the 39th Annual Conference on Neural Information Processing Systems*, pages 1–34.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 3214–3252.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *the 12th International Conference on Learning Representations*, pages 1–21.

Locutusque. 2024. Hercules v1.0.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and 1 others. 2023. The FLAN collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648.

Shayne Longpre, Robert Mahari, Anthony Chen, Naana Obeng-Marnu, Damien Sileo, William Brannon, Niklas Muennighoff, Nathan Khazam, Jad Kabbara, Kartik Perisetla, and 1 others. 2024. A large-scale audit of dataset licensing and attribution in AI. *Nature Machine Intelligence*, 6(8):975–987.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2024. #InsTag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *the 12th International Conference on Learning Representations*, pages 1–19.

Sasha Luccioni, Yacine Jernite, and Margaret Mitchell. 2021. Data measurements tool.

Run Luo, Haonan Zhang, Longze Chen, Ting-En Lin, Xiong Liu, Yuchuan Wu, Min Yang, Yongbin Li, Minzheng Wang, Pengpeng Zeng, Lianli Gao, Heng Tao Shen, Yunshui Li, Hamid Alinejad-Rokny, Xiaobo Xia, Jingkuan Song, and Fei Huang. 2025. MMEvol: Empowering multimodal large language models with evol-instruct. In *Findings of the Association for Computational Linguistics: ACL*, pages 19655–19682.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. 2024. Orca-Math: Unlocking the potential of SLMs in grade school math. *Preprint*, arXiv:2402.14830.

Zhuoshi Pan, Qizhi Pei, Yu Li, Qiyao Sun, Zinan Tang, H. Vicky Zhao, Conghui He, and Lijun Wu. 2025. REST: Stress testing large reasoning models by asking multiple problems at once. *Preprint*, arXiv:2507.10541.

Liang Pang, Kangxi Wu, Sunhao Dai, Zihao Wei, Zenghao Duan, Jia Gu, Xiang Li, Zhiyi Yin, Jun Xu, Huawei Shen, and Xueqi Cheng. 2025. Large language model sourcing: A survey. *Preprint*, arXiv:2510.10161.

Qizhi Pei, Zhuoshi Pan, Honglin Lin, Xin Gao, Yu Li, Zinan Tang, Conghui He, Rui Yan, and Lijun Wu. 2025a. ScaleDiff: Scaling difficult problems for advanced mathematical reasoning. *Preprint*, arXiv:2509.21070.

Qizhi Pei, Lijun Wu, Zhuoshi Pan, Yu Li, Honglin Lin, Chenlin Ming, Xin Gao, Conghui He, and Rui Yan. 2025b. MathFusion: Enhancing mathematical problem-solving of LLM through instruction fusion. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7400–7420.

Guilherme Penedo, Anton Lozhkov, Hynek Kydlíček, Loubna Ben Allal, Edward Beeching, Agustín Piqueres Lajarín, Quentin Gallouédec, Nathan Habib, Lewis Tunstall, and Leandro von Werra. 2025. Codeforces cots. `https://huggingface.co/datasets/open-r1/codeforces-cots`.

prithivMLmods. 2025. Open-omega-forge-1m. `https://huggingface.co/datasets/prithivMLmods/Open-Omega-Forge-1M`. Dataset.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 10776–10787.

Seyed Mahmoud Sajjadi Mohammadabadi, Burak Cem Kara, Can Eyupoglu, Can Uzay, Mehmet Serkan Tosun, and Oktay Karakuş. 2025. A survey of large language models: evolution, architectures, adaptation, benchmarking, applications, challenges, and societal implications. *Electronics*, 14(18):1–31.

Thomas Sandholm, Sayandev Mukherjee, and Bernardo A. Huberman. 2024. Randomness is all you need: Semantic traversal of problem-solution spaces with large language models. *Preprint*, arXiv:2402.06053.

Zhebei Shen, Qifan Yu, Juncheng Li, Wei Ji, Qizhi Chen, Siliang Tang, and Yueting Zhuang. 2025. Evolved-GRPO: Unlocking reasoning in LVLMs via progressive instruction evolution. In *the 39th Annual Conference on Neural Information Processing Systems*, pages 1–15.

Rizki Suwanda, Zulfahmi Syahputra, and Elvi M Zamzami. 2020. Analysis of euclidean distance and manhattan distance in the K-means algorithm for variations number of centroid K. In *Journal of Physics: Conference Series*, volume 1566, page 012058.

Teknium. 2023. OpenHermes 2.5: An open dataset of synthetic data for generalist LLM assistants.

Xiaoyu Tian, Yunjie Ji, Haotian Wang, Shuaiting Chen, Sitong Zhao, Yiping Peng, Han Zhao, and Xiangang Li. 2025. Not all correct answers are equal: Why your distillation source matters. *Preprint*, arXiv:2505.14464.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. 2025. OpenMathInstruct-2: Accelerating AI for math with massive open-source instruction data. In *the 13th International Conference on Learning Representations*, pages 1–33.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2024. SciBench: Evaluating college-level scientific problem-solving abilities of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 50622–50649.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. RedPajama: An open dataset for training large language models. *Neural Information Processing Systems Datasets and Benchmarks Track*, pages 1–31.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Tanglifu Tanglifu, Xiaowei Lv, and 1 others. 2025. Light-R1: Curriculum SFT, DPO and RL for long CoT from scratch and beyond. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 318–327.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *the 12th International Conference on Learning Representations*, pages 1–22.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2025. Magpie: Alignment data synthesis from scratch by prompting aligned LLMs with nothing. In *the 13th International Conference on Learning Representations*, pages 1–37.

Hiroshi Yoshihara, Taiki Yamaguchi, and Yuichi Inoue. 2025. A practical two-stage recipe for mathematical LLMs: Maximizing accuracy with SFT and efficiency with reinforcement learning. In *the 2nd AI for Math Workshop @ International Conference on Machine Learning*, pages 1–7.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. 2025. AnyEdit: Mastering unified high-quality image editing for any idea. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26125–26135.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2025. A survey of large language models. *Preprint*, arXiv:2303.18223.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. 2024. OpenCodeInterpreter: Integrating code generation with execution and refinement. In *Findings of the Association for Computational Linguistics: ACL*, pages 12834–12859.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2023. LIMA: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021.

Xuanhe Zhou, Junxuan He, Wei Zhou, Haodong Chen, Zirui Tang, Haoyu Zhao, Xin Tong, Guoliang Li, Youmin Chen, Jun Zhou, and 1 others. 2025. A survey of LLM× DATA. *arXiv preprint arXiv:2505.18458*.

A Implementation Details of Automatic Provenance Framework

In this section, we provide a detailed overview of the pipeline used to construct the data lineage graph. We discuss the specific models employed for each agent within our framework and list the seed datasets selected for the initial analysis.

A.1 Agent Model Configuration

To balance accuracy, efficiency, and cost, we assigned different large language models to specific agents based on their performance characteristics.

Sourcing Agent. This agent is responsible for accurately identifying entry points for data information from repository README files. To minimize hallucination risks and ensure precise extraction, we utilized `GPT-5.1`. Its strong instruction-following capability ensures that the initial retrieval of metadata is reliable.

Extracting Agent. This agent visits the identified sources and summarizes the information into a specified format. Given the high volume of text processing required, we selected `Gemini-2.5-Flash`. Its high processing speed allows for efficient large-scale information scanning and summarization without increased latency.

Tracing Agent. After information summary, this agent extracts the specific source-target relationships. Similar to the Sourcing Agent, precision is critical here to avoid false lineages. Therefore, we again employed `GPT-5.1` to leverage its low hallucination rate for accurate relationship extraction.

Aggregation Agent. This agent handles the standardization of dataset names and the merging of sources. We utilized `Gemini-2.5-Pro` for this task. Its strong reasoning capabilities and support for web retrieval allow it to resolve ambiguous dataset names by verifying them against online resources. It effectively consolidates dispersed information into unified nodes.

A.2 Seed Data Selection

We curated a set of 83 seed datasets to serve as the starting point for our lineage analysis. These datasets were selected based on three criteria: download volume, community engagement (likes or stars), and the reported performance of downstream models trained on them. All selected datasets are verifiable and retrievable on HuggingFace. The complete list is provided in Table 5.

A.3 Relation Types in Data Lineage

The relation type R in our lineage graph characterizes the specific derivation methods between datasets. We identify five primary categories:

- **Semantic Evolution**: Reformulating or enhancing the original question while allowing controlled semantic variation.
- **CoT Distillation**: Keeping the question unchanged while using a stronger teacher model to generate long CoT responses.
- **Synthetic Generation**: Using upstream data as seeds to generate new question-answer pairs through LLM generalization.
- **Structured Fusion**: Concatenating or combining data from multiple sources for composite reasoning.
- **Direct Inclusion/Subset**: Including upstream data as-is as part of the new dataset.

Role in graph construction. The tracing agent infers relation type R by analyzing method descriptions in documentation (papers, READMEs, etc.) and stores it as an edge attribute in the graph. It should be noted that some datasets have vague descriptions (e.g., only mentioning "based on XX" without specifying the processing method). In such cases, the agent makes the best inference based on contextual semantics or labels it as the default "Direct Inclusion" relation.

Storing R as an edge attribute serves two key purposes: (1) it enhances graph semantic richness by enabling the graph to answer not only "dataset A is derived from dataset B" but also "through what method," and (2) it supports downstream applications such as data deduplication, contamination tracking, and quality assessment by providing fine-grained signals about derivation patterns.

B Topological Structure of the Lineage Graph

Based on the extensive lineage graph constructed, we provide a detailed statistical analysis of the ecosystem's topological properties. We focus on three key dimensions to characterize the roles and evolutionary patterns of different datasets: reuse

Seed Datasets List		
m-a-p/CodeFeedback-Filtered-Instruction	garage-bAInd/Open-Platypus	amphora/QwQ-LongCoT-130K
open-thoughts/OpenThoughts3-1.2M	WizardLMTeam/WizardLM_evol_instruct_V2_196k	ajibawa-2023/Code-74k-ShareGPT
a-m-team/AM-Thinking-v1-Distilled	gretelai/gretel-text-to-python	QuixiAI/dolphin
microsoft/rStar-Coder	TokenBender/code_instructions_122k	EricLu/SCP-116K
OpenCoder-LLM/opc-sft-stage2	sequelbox/Raiden-DeepSeek-R1	GAIR/o1-journey
Locutusque/hercules-v1.0	Magpie-Align/Magpie-Reasoning-V2-250K	miromind-ai/MiroMind-M1-SFT-719K
theblackcat102/evol-codealpaca-v1	PrimeIntellect/SYNTHETIC-2-SFT-verified	zwhe99/DeepMath-103K
ajibawa-2023/Code-290k-ShareGPT	MegaScience/TextbookReasoning	open-r1/OpenR1-Math-220k
KodCode/KodCode-V1	qihoo360/Light-R1-SFTData	dyyyyyyyy/ScaleQuest-Math
m-a-p/Code-Feedback	Magpie-Align/Magpie-Reasoning-V2-250K	AI-MO/NuminaMath-1.5
amphora/QwQ-LongCoT-130K-2	Magpie-Align/Magpie-Reasoning-V1-150K	AI-MO/NuminaMath-CoT
allenai/tulu-3-sft-mixture	O1-OPEN/OpenO1-SFT	PawanKrd/math-gpt-4o-200k
OpenCoder-LLM/opc-sft-stage1	RabotniKuma/Fast-Math-R1-SFT	bespokelabs/Bespoke-Stratos-17k
microsoft/EpiCoder-func-380k	SkunkworksAI/reasoning-0.01	hkust-nlp/dart-Math-hard
MegaScience/MegaScience	tatsu-lab/alpaca	TIGER-Lab/WebInstruct-CFT
microsoft/orca-agentinstruct-1M-v1	vicgalle/alpaca-gpt4	rubenroy/GammaCorpus-CoT-Math-170k
alibaba-pai/OmniThought-0528	allenai/tulu-3-sft-personas-math	allenai/tulu-3-sft-personas-algebra
ajibawa-2023/Python-Code-23k-ShareGPT	QizhiPei/MathFusionQA	TIGER-Lab/MATH-plus
likaixin/InstructCoder	whynlp/gsm8k-aug	GAIR/LIMO
Mxode/Magpie-Pro-10K-GPT4o-mini	nvidia/OpenMathInstruct-2	microsoft/orca-math-word-problems-200k
efficientscaling/Z1-Code-Reasoning-107K	ajibawa-2023/Maths-College	ServiceNow-AI/R1-Distill-SFT
teknium/OpenHermes-2.5	agentica-org/DeepScaleR-Preview-Dataset	databricks/databricks-dolly-15k
ise-uiuc/Magicoder-OSS-Instruct-75K	gretelai/synthetic_text_to_sql	open-r1/codeforces-cots
nickrosh/Evol-Instruct-Code-80k-v1	QizhiPei/ScaleDiff-Math	LHL3341/Caco-1.3M
WizardLMTeam/WizardLM_evol_instruct_70k	gretelai/gretel-math-gsm8k-v1	prithivMLmods/Open-Omega-Forge-1M
open-thoughts/OpenThoughts-114k	OpenCoder-LLM/opc-sft-stage1	hakurei/open-instruct-v1
GAIR/lima	allenai/omega-explorative	openbmb/UltraInteract_sft
allenai/tulu-3-sft-personas-code	bigcode/self-oss-instruct-sc2-exec-filter-50k	

Table 5: Full list of the 83 seed datasets used for lineage analysis. The datasets are sourced from HuggingFace and cover Math, Code, General, and Science domains.

rate (measured by out-degree), information aggregation (measured by in-degree), and evolutionary depth. The specific statistics for these dimensions are presented in Tables 6, 7, and 8, respectively.

C Source Intersection Details

As discussed in Section 4.2, we conducted a rigorous intersection analysis using a strict matching protocol. We calculated hash values based on the complete (`instruction`, `input`, `output`) triplets to identify exact duplicates. Even under this strict criterion, we detected significant redundancy across multiple datasets.

Table 9 details the specific upstream intersection paths identified in our analysis. These paths reveal how certain datasets inadvertently incorporate large portions of upstream sources. We recommend that data curators adopt this verification method to detect hidden upstream intersections. This practice is essential for preventing the redundant selection of identical data sources when constructing a new training pool.

D Data Contamination Details

Expanding on the analysis in Section 4.3, this section details the downstream propagation of data contamination. After obtaining lineage relationships, we perform strict exact matching of benchmark samples (`instruction`, `input`) against downstream training data to measure actual contamination rates. We broaden the scope from the main text to include comprehensive statistics for five benchmarks, incorporating LiveCodeBench and TruthfulQA.

Special attention is required for LiveCodeBench due to its chronological update mechanism. Our analysis reveals that datasets such as `a-m-team/AM-Thinking-v1-Distilled` and `agentica-org/DeepCoder-Preview-Dataset` inadvertently incorporated test samples from LiveCodeBench v5[3]. This exposes the critical risks associated with temporal benchmarks when strict version control and temporal cutoffs are neglected.

Table 10 provides granular contamination statistics, reporting both the exact count and percentage of leaked samples. Additionally, Table 12 traces the specific lineage pathways through which

[3] https://livecodebench.github.io/leaderboard_v5.html

Domain	Rank	Dataset Name	Out-Degree	Release Date
Math	1	EleutherAI/hendrycks_math	19	2021-03
	2	openai/gsm8k	14	2021-10
	3	AI-MO/NuminaMath-CoT	13	2024-07
	4	open-r1/OpenR1-Math-220k	6	2025-02
	5	meta-math/MetaMathQA	6	2023-09
Code	1	BAAI/TACO	11	2023-12
	2	codeparrot/apps	10	2021-05
	3	deepmind/code_contests	9	2021-05
	4	sahil2801/CodeAlpaca-20k	6	2023-03
	5	ise-uiuc/Magicoder-Evol-Instruct-110K	5	2023-12
General	1	Open-Orca/FLAN	7	2021-09
	2	tatsu-lab/alpaca	6	2023-03
	3	anon8231489123/ShareGPT_Vicuna_unfiltered	5	2023-04
	4	teknium/GPTeacher-General-Instruct	4	2023-04
	5	wikimedia/wikipedia	4	2022-03
Science	1	open-thoughts/OpenThoughts-114k	3	2025-01
	2	camel-ai/chemistry	3	2023-03
	3	camel-ai/physics	3	2023-03
	4	camel-ai/biology	3	2023-03
	5	nvidia/Llama-Nemotron-Post-Training-Dataset	3	2025-03

Table 6: Top 5 most reused datasets by domain (ranked by out-degree).

Rank	Dataset Name	Domain	In-Degree	Release Date
1	HuggingFaceFW/fineweb	General	111	2021-09
2	bigscience/xP3	Code, General	79	2022-10
3	CohereLabs/aya_collection	General	70	2024-01
4	bigscience/P3	General	54	2021-10
5	allenai/lila	Math	20	2023-02
6	izumi-lab/llm-japanese-dataset-vanilla	General	20	2023-05
7	a-m-team/AM-Thinking-v1-Distilled	Math, Code, Science, General	19	2025-05
8	teknium/OpenHermes-2.5	Math, Code, Science, General	19	2023-11
9	izumi-lab/llm-japanese-dataset	General	19	2023-04
10	allenai/tulu-3-sft-mixture	Math, Code, Science, General	18	2024-11

Table 7: Top 10 datasets with highest global in-degree.

contamination enters the ecosystem. These findings empirically confirm that contamination in upstream sources inevitably cascades into downstream derivatives.

Consequently, we strongly advocate for the strict exclusion of benchmark samples from training corpora. Even if a model is not intended for evaluation on a specific benchmark, retaining these samples contaminates the shared data lineage, compromising future research. We emphasize that rigorous decontamination protocols are prerequisite to ensuring the validity of generalization assessments and preventing inflated evaluation metrics.

E Diversity Optimization via Leaf Nodes

Provenance-based sampling workflow. In our provenance-based sampling strategy, we leverage leaf nodes ($d_{in} = 0$) as **upstream knowledge anchors** to construct a diversity-oriented dataset. By anchoring sampling on leaf nodes, we reduce structural redundancy at the source, as these nodes represent independent original knowledge sources. The complete workflow proceeds as follows:

- **Leaf node selection**: Filter all leaf nodes from the lineage graph (212 unique datasets; the per-domain leaf counts in Table 1 sum to a larger value because some leaves span multiple domains), and rank them by d_{out}, excluding nodes with zero downstream usage.
- **Domain filtering**: Retain domains commonly targeted in post-training (Math, Code, Science, etc.), filter out niche domains, and exclude non-QA format data (e.g., Common Crawl, Wikipedia), resulting in 31 datasets.
- **Format unification**: Convert all data to Alpaca format, yielding approximately 8.7M samples.

Rank	Dataset Name	Domain	Depth	Release Date
1	alibaba-pai/OmniThought	Code, General, Math	9	2025-05
2	allenai/tulu-3-sft-mixture	Math, Code, Science, General	8	2024-11
3	zwhe99/DeepMath-103K	Math	8	2025-04
4	CohereForAI/aya_dataset	General	7	2024-01
5	open-thoughts/OpenThoughts2-1M	Science, Code, General, Math	7	2025-04

Table 8: Top 5 datasets with highest evolutionary depth.

- **Initial filtering**: Remove overly long/short samples and non-English data.
- **Two-stage deduplication**: Apply strict exact Q-matching deduplication, followed by MinHash deduplication (128 hash permutations, threshold 0.7, N-gram 8), producing 570K high-quality instruction samples.

The subsequent deduplication further eliminates sample-level overlaps between different leaf nodes. Table 11 provides the complete list of the 31 core datasets utilized in our curation process after domain filtering.

Diversity Metric Calculations. To rigorously evaluate the semantic span of the curated data and baseline data, we employed two complementary metrics. Let $X = \{x_1, x_2, \ldots, x_N\}$ be the set of embeddings for the instructions in the dataset, where N is the sample size, computed using `Qwen/Qwen3-Embedding-8B` (Zhang et al., 2025). We fix the embedding dimensionality to 4096 for all samples.

(1) Vendi Score (Intrinsic Diversity) The Vendi Score (Friedman and Dieng, 2022) interprets diversity as the effective number of unique semantic clusters. It is calculated based on the eigenvalues of the kernel matrix K, where $K_{ij} = k(x_i, x_j)$ represents the similarity between samples (we use the cosine similarity kernel). The Vendi Score is defined as the exponential of the Shannon entropy of the eigenvalues:

$$\text{Vendi}(X) = \exp\left(-\sum_{i=1}^{N} \lambda_i \ln \lambda_i\right), \quad (1)$$

where $\lambda_1, \ldots, \lambda_N$ are the normalized eigenvalues of the matrix K/N. A higher Vendi Score indicates a dataset with a larger number of effective independent modes. In practice, we compute Vendi using the reference implementation provided by the original authors.[4]

(2) Centroid Distance (Geometric Dispersion). This metric measures the spatial spread of the data points in the high-dimensional embedding space. We first compute the global centroid μ of the dataset:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i. \quad (2)$$

The Centroid Distance is then defined as the complement of the average cosine similarity between each sample x_i and the centroid μ:

$$\text{Dist}_{cent}(X) = 1 - \frac{1}{N}\sum_{i=1}^{N} \frac{x_i \cdot \mu}{\|x_i\|\|\mu\|}. \quad (3)$$

A higher Centroid Distance implies that the samples are widely dispersed around the center, covering a broader semantic region rather than clustering tightly around a single topic.

[4] https://github.com/vertaix/Vendi-Score

Target Dataset	Complete Evolutionary Paths	Intersection Point
CodeFeedback-Filtered-Instruction	**Path 1:** m-a-p/CodeFeedback-Filtered-Instruction → ise-uiuc/Magicoder-Evol-Instruct-110K → theblackcat102/evol-codealpaca-v1 → HuggingFaceH4/CodeAlpaca_20K → sahil2801/CodeAlpaca-20k **Path 2:** m-a-p/CodeFeedback-Filtered-Instruction → nickrosh/Evol-Instruct-Code-80k-v1 → sahil2801/CodeAlpaca-20k	sahil2801/CodeAlpaca-20k
Fast-Math-R1-SFT	**Path 1:** RabotniKuma/Fast-Math-R1-SFT → open-r1/OpenR1-Math-220k **Path 2:** RabotniKuma/Fast-Math-R1-SFT → qihoo360/Light-R1-SFTData → open-r1/OpenR1-Math-220k	open-r1/OpenR1-Math-220k
Open-Omega-Forge-1M	**Path 1:** prithivMLmods/Open-Omega-Forge-1M → nvidia/OpenCodeReasoning → codeparrot/apps **Path 2:** prithivMLmods/Open-Omega-Forge-1M → nvidia/OpenMathReasoning → nvidia/Llama-Nemotron-Post-Training-Dataset → codeparrot/apps	codeparrot/apps
Light-R1-SFTData	**Path 1:** qihoo360/Light-R1-SFTData → open-r1/OpenR1-Math-220k → AI-MO/NuminaMath-CoT **Path 2:** qihoo360/Light-R1-SFTData → open-thoughts/OpenThoughts-114k → AI-MO/NuminaMath-CoT	AI-MO/NuminaMath-CoT
	Path 3: qihoo360/Light-R1-SFTData → nvidia/OpenMathInstruct-2 → EleutherAI/hendrycks_math **Path 4:** qihoo360/Light-R1-SFTData → GAIR/LIMO → EleutherAI/hendrycks_math	EleutherAI/hendrycks_math
OpenCodeReasoning	**Path 1:** nvidia/OpenCodeReasoning → BAAI/TACO → codeparrot/apps **Path 2:** nvidia/OpenCodeReasoning → codeparrot/apps	codeparrot/apps
open-instruct-v1	**Path 1:** hakurei/open-instruct-v1 → tatsu-lab/alpaca → yizhongw/self_instruct **Path 2:** hakurei/open-instruct-v1 → yizhongw/self_instruct	yizhongw/self_instruct

Table 9: Full lineage paths: tracing the complete evolution from target to source (selected datasets).

Training Dataset	Contamination Ratio
Target Benchmark: **Omni-MATH**	
Skywork/Skywork-OR1-RL-Data	96.80% (4265/4406)
agentica-org/DeepScaleR-Preview-Dataset	79.48% (3502/4406)
SynthLabsAI/Big-Math-RL-Verified	57.97% (2554/4406)
LHL3341/Caco-1.3M	37.95% (1672/4406)
a-m-team/AM-Thinking-v1-Distilled	28.94% (1275/4406)
PrimeIntellect/SYNTHETIC-2-SFT-verified	23.01% (1014/4406)
RabotniKuma/Fast-Math-R1-SFT	4.86% (214/4406)
qihoo360/Light-R1-SFTData	2.77% (122/4406)
Target Benchmark: **TheoremQA**	
garage-bAInd/Open-Platypus	70.59% (564/799)
TIGER-Lab/MathInstruct	66.96% (535/799)
teknium/OpenHermes-2.5	62.33% (498/799)
a-m-team/AM-Thinking-v1-Distilled	46.43% (371/799)
open-thoughts/OpenThoughts2-1M	8.89% (71/799)
alibaba-pai/OmniThought-0528	4.38% (35/799)
Target Benchmark: **LiveCodeBench**	
agentica-org/DeepCoder-Preview-Dataset	88.12% (89/101)
a-m-team/AM-Thinking-v1-Distilled	44.55% (45/101)
Target Benchmark: **TruthfulQA**	
openbmb/UltraFeedback	99.27% (811/817)
Target Benchmark: **SciBench**	
garage-bAInd/Open-Platypus	76.23% (526/690)
teknium/OpenHermes-2.5	66.09% (456/690)

Table 10: Data contamination analysis: leakage ratios of training datasets on various benchmarks.

Seed Datasets List	
sahil2801/CodeAlpaca-20k	deepmind/aqua_rat
glaiveai/glaive-code-assistant-v3	ajibawa-2023/Python-Code-23k-ShareGPT
allenai/sciq	mlfoundations-dev/stackexchange_physics
ise-uiuc/Magicoder-OSS-Instruct-75K	allenai/qasc
justus27/reasoning-gym-genesys	camel-ai/biology
autoprogrammer/nemotron_science_lf_filtered	jondurbin/airoboros-2.1
teknium/GPT4-LLM-Cleaned	PrimeIntellect/synthetic-code-understanding
mlfoundations-dev/stackexchange_codegolf	camel-ai/physics
PrimeIntellect/real-world-swe-problems	HARP(Human Annotated Reasoning Problems)
MatrixStudio/Codeforces-Python-Submission	PrimeIntellect/stackexchange-question-answering
hoanganhpham/openr1_hard	sopen-r1/codeforces
avaliev/ChemistryQA	nvidia/OpenScience
camel-ai/chemistry	hivaze/LOGIC-701
LooksJuicy/ruozhiba	stanfordnlp/web_questions
Multilingual-Multimodal-NLP/McEval-Instruct	di-zhang-fdu/AIME_1983_2024
Magpie-Align/Magpie-Reasoning-V2-250K-CoT-Llama3	

Table 11: List of the 31 Leaf Nodes ($d_{in} = 0$) selected as upstream knowledge anchors. These datasets formed the initial pool for our provenance-based sampling strategy.

Training Dataset	Evolutionary Path (Source → Target)
Target Benchmark: **Omni-MATH**	
GAIR/LIMO	KbsdJames/Omni-MATH → agentica-org/DeepScaleR-Preview-Dataset → GAIR/LIMO
LHL3341/Caco-1.3M	**1.** KbsdJames/Omni-MATH → agentica-org/DeepScaleR-Preview-Dataset → LHL3341/Caco-1.3M **2.** KbsdJames/Omni-MATH → SynthLabsAI/Big-Math-RL-Verified → LHL3341/Caco-1.3M
PrimeIntellect/ SYNTHETIC-2-SFT-verified	**1.** KbsdJames/Omni-MATH → Skywork/Skywork-OR1-RL-Data → PrimeIntellect/SYNTHETIC-2-SFT-verified **2.** KbsdJames/Omni-MATH → agentica-org/DeepScaleR-Preview-Dataset → Skywork/Skywork-OR1-RL-Data → PrimeIntellect/SYNTHETIC-2-SFT-verified
RabotniKuma/ Fast-Math-R1-SFT	**1.** KbsdJames/Omni-MATH → qihoo360/Light-R1-SFTData → RabotniKuma/Fast-Math-R1-SFT **2.** KbsdJames/Omni-MATH → agentica-org/DeepScaleR-Preview-Dataset → GAIR/LIMO → qihoo360/Light-R1-SFTData → RabotniKuma/Fast-Math-R1-SFT
Skywork/ Skywork-OR1-RL-Data	**1.** KbsdJames/Omni-MATH → Skywork/Skywork-OR1-RL-Data **2.** KbsdJames/Omni-MATH → agentica-org/DeepScaleR-Preview-Dataset → Skywork/Skywork-OR1-RL-Data
SynthLabsAI/ Big-Math-RL-Verified	KbsdJames/Omni-MATH → SynthLabsAI/Big-Math-RL-Verified
a-m-team/ AM-Thinking-v1-Distilled	KbsdJames/Omni-MATH → SynthLabsAI/Big-Math-RL-Verified → a-m-team/AM-Thinking-v1-Distilled
agentica-org/ DeepScaleR-Preview-Dataset	KbsdJames/Omni-MATH → agentica-org/DeepScaleR-Preview-Dataset
qihoo360/ Light-R1-SFTData	**1.** KbsdJames/Omni-MATH → qihoo360/Light-R1-SFTData **2.** KbsdJames/Omni-MATH → agentica-org/DeepScaleR-Preview-Dataset → GAIR/LIMO → qihoo360/Light-R1-SFTData
Target Benchmark: **TruthfulQA**	
openbmb/UltraFeedback	truthfulqa/truthful_qa → openbmb/UltraFeedback
Target Benchmark: **LiveCodeBench**	
a-m-team/ AM-Thinking-v1-Distilled	PrimeIntellect/LiveCodeBench-v5 → agentica-org/DeepCoder-Preview-Dataset → a-m-team/AM-Thinking-v1-Distilled
agentica-org/ DeepCoder-Preview-Dataset	PrimeIntellect/LiveCodeBench-v5 → agentica-org/DeepCoder-Preview-Dataset
Target Benchmark: **TheoremQA**	
QizhiPei/ScaleDiff-Math	TIGER-Lab/TheoremQA → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → TIGER-Lab/WebInstructSub → zwhe99/DeepMath-103K → QizhiPei/ScaleDiff-Math
TIGER-Lab/MathInstruct	TIGER-Lab/TheoremQA → TIGER-Lab/MathInstruct
TIGER-Lab/WebInstruct-CFT	TIGER-Lab/TheoremQA → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → TIGER-Lab/WebInstructSub → TIGER-Lab/WebInstruct-CFT
TIGER-Lab/WebInstructSub	TIGER-Lab/TheoremQA → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → TIGER-Lab/WebInstructSub
a-m-team/ AM-Thinking-v1-Distilled	TIGER-Lab/TheoremQA → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → a-m-team/AM-Thinking-v1-Distilled
alibaba-pai/OmniThought	**1.** TIGER-Lab/TheoremQA → TIGER-Lab/MathInstruct → open-thoughts/OpenThoughts2-1M → alibaba-pai/OmniThought **2.** TIGER-Lab/TheoremQA → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → TIGER-Lab/WebInstructSub → zwhe99/DeepMath-103K → alibaba-pai/OmniThought

Continued on next page...

Training Dataset	Evolutionary Path (Source → Target)
alibaba-pai/OmniThought-0528	**1.** TIGER-Lab/TheoremQA → TIGER-Lab/MathInstruct → open-thoughts/OpenThoughts2-1M → alibaba-pai/OmniThought → alibaba-pai/OmniThought-0528 **2.** TIGER-Lab/TheoremQA → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → TIGER-Lab/WebInstructSub → zwhe99/DeepMath-103K → alibaba-pai/OmniThought → alibaba-pai/OmniThought-0528
garage-bAInd/Open-Platypus	TIGER-Lab/TheoremQA → garage-bAInd/Open-Platypus
open-thoughts/OpenThoughts2-1M	TIGER-Lab/TheoremQA → TIGER-Lab/MathInstruct → open-thoughts/OpenThoughts2-1M
teknium/OpenHermes-2.5	TIGER-Lab/TheoremQA → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5
zwhe99/DeepMath-103K	TIGER-Lab/TheoremQA → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → TIGER-Lab/WebInstructSub → zwhe99/DeepMath-103K
***Target Benchmark:* SciBench**	
QizhiPei/ScaleDiff-Math	xw27/scibench → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → TIGER-Lab/WebInstructSub → zwhe99/DeepMath-103K → QizhiPei/ScaleDiff-Math
TIGER-Lab/WebInstruct-CFT	xw27/scibench → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → TIGER-Lab/WebInstructSub → TIGER-Lab/WebInstruct-CFT
TIGER-Lab/WebInstructSub	xw27/scibench → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → TIGER-Lab/WebInstructSub
a-m-team/AM-Thinking-v1-Distilled	xw27/scibench → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → a-m-team/AM-Thinking-v1-Distilled
alibaba-pai/OmniThought	xw27/scibench → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → TIGER-Lab/WebInstructSub → zwhe99/DeepMath-103K → alibaba-pai/OmniThought
alibaba-pai/OmniThought-0528	xw27/scibench → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → TIGER-Lab/WebInstructSub → zwhe99/DeepMath-103K → alibaba-pai/OmniThought → alibaba-pai/OmniThought-0528
garage-bAInd/Open-Platypus	xw27/scibench → garage-bAInd/Open-Platypus
teknium/OpenHermes-2.5	xw27/scibench → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5
zwhe99/DeepMath-103K	xw27/scibench → garage-bAInd/Open-Platypus → teknium/OpenHermes-2.5 → TIGER-Lab/WebInstructSub → zwhe99/DeepMath-103K

Table 12: Data lineage analysis: tracking the usage of key benchmarks in downstream training datasets.