DataDignity: Training Data Attribution for Large Language Models

Xiaomin Li* Microsoft

Andrzej Banburski-Fahey Microsoft

Jaron Lanier Microsoft

Abstract

Auditing language-model outputs often requires more than judging correctness: an auditor may need to know which source document most likely supports the knowledge expressed in a response. We study this problem as *pinpoint provenance*: given a prompt, a target-model response, and a candidate corpus, rank the documents that best support the response. We introduce **FAKEWIKI**, a controlled benchmark of 3,537 fabricated Wikipedia-style articles designed to preserve ground-truth provenance while weakening lexical shortcuts. Each evaluated target LLM is explicitly continued-pretrained on the FAKEWIKI text corpus before response collection, while the QA probes used for attribution evaluation are held out from target-model training. FAKEWIKI includes short QA probes, source-preserving paraphrases, retro-generated variants, hard anti-documents that remain topically similar while removing answer-critical facts, and five query conditions: clean prompting plus four jailbreak-inspired transformations, obfuscation, role-play, noise injection, and indirect prompting. We evaluate eleven lexical and semantic retrieval baselines, a training-free activation-steering retrieval-fusion method **STEERFUSE**, and a supervised contrastive provenance ranker **SCORINGMODEL**. SCORINGMODEL maps response and document features into a shared space and is trained with InfoNCE using in-batch, retrieval-mined, and anti-document negatives. Across nine open-weight instruction-tuned LLMs and five query conditions, SCORINGMODEL improves mean Recall@10 from 37.3 for the strongest retrieval baseline to 52.2, without inference-time fusion, and wins 41/45 model-by-condition cells. STEERFUSE beats the strongest retrieval baseline in most cells while requiring no supervised training, showing that activation-space evidence can complement text retrieval. On the jailbreak-inspired transformed queries, SCORINGMODEL improves Recall@10 by 13.2 points on average over the best baseline, with the largest gains on larger target models. Overall, our work shows that robust training data attribution requires evaluation settings that separate true answer support from topical or lexical resemblance.

1 Introduction

Large language models increasingly mediate factual, scientific, legal, and safety-relevant information. When a model produces a response, users may need to know not only whether it is correct, but also where it came from: which source document supplied the relevant fact, whether a questionable output depends on a particular source, or whether a data intervention removed the intended provenance path. These questions arise in copyright audits, misinformation forensics, safety debugging, and dataset curation, and are not fully answered by standard evaluation or influence-style methods [Han and Tsvetkov, 2021, Li et al., 2026, Zhang et al., 2024, Akyürek et al., 2022, Park et al., 2023, Barshan et al., 2020].

*Correspondence: xiaominli@microsoft.com.

Preprint.

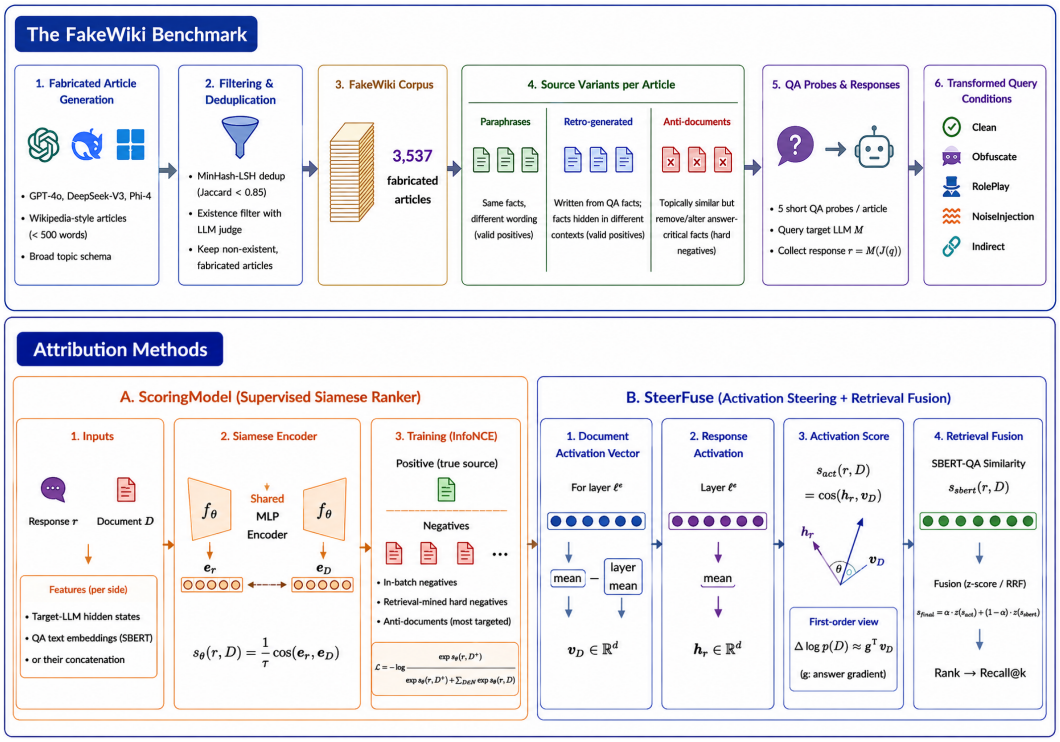


Figure 1: Overview of DATADIGNITY. Top: FAKEWIKI constructs fabricated source documents, variants, anti-documents, and transformed queries. Bottom: SCORINGMODEL learns a supervised provenance score, while STEERFUSE fuses activation-space evidence with SBERT retrieval.

We study this problem as *pinpoint provenance*. Given a prompt x, a target-model response y, and a candidate corpus $\mathcal{D} = \{D_j\}_{j=1}^{N}$, the goal is to return a short ranked list of documents that likely support the knowledge expressed in y. This is an operational retrieval problem: an auditor should inspect a small set of candidate sources rather than search through an entire corpus. It is harder than ordinary semantic retrieval because the answer may be short, paraphrased, grounded in a small fact buried in a longer document, or elicited through a prompt transformation.

A central challenge is that many provenance evaluations make attribution too easy through surface overlap. If the source document, question, and response share rare names or distinctive phrases, lexical methods such as MinHash [Broder, 1997], and even generic dense retrievers such as SBERT [Reimers and Gurevych, 2019], Contriever [Izacard et al., 2022], and BGE [Xiao et al., 2024], can appear effective without demonstrating robust source attribution. Such methods may fail when provenance matters most: under paraphrase, obfuscation, indirect questioning, role-play, or irrelevant context injection. This motivates a benchmark in which the true source is known by construction, but the evaluation deliberately removes the easy paths from response wording back to document identity.

We introduce FAKEWIKI, a benchmark designed to preserve ground-truth provenance while weakening such shortcuts. It contains 3,537 fabricated Wikipedia-style articles with short QA probes, source-preserving variants, and hard anti-documents that preserve topical similarity while removing answer-critical facts. To make this a training-data attribution setting, each target LLM is continued-pretrained on FAKEWIKI document text, while QA probes are held out and used only to elicit responses whose provenance should point back to the training documents. We evaluate attribution under clean prompts and four transformed conditions: `Obfuscate`, `RolePlay`, `NoiseInjection`, and `Indirect`, testing whether attribution survives when lexical and semantic cues become less reliable. Figure 1 summarizes the benchmark and attribution pipeline.[2]

Our main attribution method, SCORINGMODEL, is a supervised Siamese provenance ranker. It maps response-side and document-side features into a shared embedding space and trains with a contrastive InfoNCE objective [Oord et al., 2018] over in-batch negatives, retrieval-mined hard negatives, and

[2]Data and code are available at `https://anonymous.4open.science/r/Submission-DataDignity-E263`.

curated anti-documents. These anti-documents force the model to distinguish documents that merely resemble the response from documents that actually support it. At inference time, each candidate document is scored by this learned compatibility function.

We also study STEERFUSE, a training-free activation-steering retrieval-fusion method inspired by representation-level interventions in language models [Subramani et al., 2022, Turner et al., 2023, Panickssery et al., 2023, Zou et al., 2023, Li et al., 2023]. It asks which candidate document provides the largest internal evidence boost toward the observed response, using cached document activation directions and an efficient response-side proxy instead of patched forward passes. The resulting activation-space score is fused with SBERT retrieval to test whether model-internal evidence complements text similarity under transformed prompts.

The main result is that clean retrieval substantially understates the difficulty of robust provenance. Both proposed attribution methods improve over standard retrieval: the training-free STEERFUSE method beats the strongest of eleven retrieval baselines in 32/45 model-by-query-condition cells, while SCORINGMODEL wins 41/45 cells. Averaged across all models and query conditions, STEERFUSE improves mean Recall@10 from 37.3 to 42.3, and SCORINGMODEL further improves it to 52.2 without inference-time fusion. On transformed queries, SCORINGMODEL improves Recall@10 by 13.2 points on average over the best baseline, with especially large gains of +26.9 on `Llama-3.1-8B` and +20.0 on `Qwen3-8B`. Recall@1 and Recall@5 show the same pattern under stricter cutoffs, especially for the larger target models. These results suggest that robust provenance evaluation should not stop at clean lexical or semantic retrieval: training-free activation evidence can improve retrieval in many settings, and supervised attribution with hard negatives can recover stronger source-support signals missed by generic similarity.

Our contributions are:

- We formulate robust pinpoint provenance as a source-attribution task that evaluates whether methods can distinguish true answer support from topical or lexical resemblance.
- We introduce FAKEWIKI, a benchmark with ground-truth source documents, short QA probes, source-preserving variants, hard anti-documents, and transformed query conditions.
- We propose SCORINGMODEL, a supervised contrastive provenance scorer trained with hard negatives and evaluated without inference-time retrieval fusion.
- We provide a broad empirical study across nine open-weight instruction-tuned LLMs, five query conditions, eleven retrieval baselines, STEERFUSE, and SCORINGMODEL, with additional per-model, seed, Recall@1, Recall@5, and ablation analyses in the appendix.

2 Related Work

Training data attribution and source retrieval. Training data attribution asks which examples or documents are associated with model behavior. Influence-style methods estimate effects on predictions or losses through gradients, checkpoints, approximations, or scalable surrogates [Pruthi et al., 2020, Han and Tsvetkov, 2021, Barshan et al., 2020, Park et al., 2023, Kwon et al., 2023], but address a complementary causal question about training dynamics. We study an operational provenance task: given a candidate corpus and generated response, rank inspectable source documents. This is closest to retrieval-based source tracing, where MinHash captures lexical overlap [Broder, 1997], while SBERT, Contriever, BGE, and finetuned embeddings capture semantic similarity [Reimers and Gurevych, 2019, Izacard et al., 2022, Xiao et al., 2024, Rajani et al., 2019, Fotouhi et al., 2024]. Related work also studies source-aware factual tracing and contrastive attribution embeddings [Akyürek et al., 2022, Khalifa et al., 2024, Wang et al., 2024]. We evaluate retrieval-based provenance under anti-shortcut conditions that separate answer support from topical or lexical resemblance.

Activation-space evidence. Activation-space methods use internal hidden states to interpret or alter model behavior. Prior work has extracted latent steering vectors [Subramani et al., 2022], added activation directions at inference time [Turner et al., 2023, Panickssery et al., 2023], and used hidden representations for monitoring, control, truthfulness, or latent-knowledge readouts [Zou et al., 2023, Li et al., 2023, Burns et al., 2023]. We build on this perspective for provenance: a candidate document may provide internal evidence for a response even when its wording is not close to the generated text. STEERFUSE tests this idea by comparing document-induced activation directions with response

Component	What it contains	What it tests
Fabricated articles	3,537 Wikipedia-style documents about non-real entities and concepts	Controlled provenance after explicit target-model exposure, without relying on real-world pretraining knowledge
QA probes	Five short question-answer probes per document	Whether attribution works when responses contain only sparse source evidence
Source variants	Paraphrases, retro-generated documents, and anti-documents	Whether methods distinguish true answer support from topical or lexical similarity
Query conditions	`Clean`, `Obfuscate`, `RolePlay`, `NoiseInjection`, and `Indirect`	Whether provenance survives prompt transformations that change surface cues

Table 1: Design of the FAKEWIKI benchmark. Each component is intended to weaken a different shortcut that ordinary retrieval methods might exploit.

representations and fusing the resulting signal with SBERT retrieval. We treat this activation-space evidence as complementary to text retrieval rather than as a replacement for it.

3 The FAKEWIKI Benchmark

A provenance benchmark should provide ground-truth sources without making attribution solvable by rare names or copied phrases. FAKEWIKI addresses this tension with fabricated Wikipedia-style articles, source-preserving variants, anti-documents, and transformed prompts. Table 1 summarizes the benchmark components. Together, they weaken wording overlap, vary factual context, preserve hard topical distractors, and disrupt prompt-response surface form.

3.1 Document Corpus

FAKEWIKI contains 3,537 fabricated Wikipedia-style articles. We build the corpus in three stages:

1. **Generate.** `GPT-4o` [Hurst et al., 2024], `DeepSeek-V3` [Liu et al., 2024], and `Phi-4` [Abdin et al., 2024] write short, internally consistent encyclopedic articles about entities or concepts that should not exist in the real world.
2. **Diversify.** We sample across fictional people, places, artifacts, events, organizations, and technical concepts so that the corpus is not dominated by a single template.
3. **Deduplicate and filter.** We remove near-duplicates with MinHash-LSH at a Jaccard threshold of 0.85 [Broder, 1997], then use an LLM existence filter to discard titles judged likely to correspond to real public entities, events, or concepts.

Surviving articles are assigned stable document identifiers and form a controlled fabricated corpus that target models should not know from ordinary pretraining. We then inject this corpus into each target model through continued pretraining, so the attribution task asks whether a method can recover which injected training document supports a later response.

3.2 Target-Model Exposure and Evaluation Split

For every target LLM in Section 5, we start from the public instruction-tuned checkpoint and continue pretrain it on FAKEWIKI text with a causal language-modeling objective. The corpus contains original articles and constructed variants, but not QA probes, reference answers, or transformed queries. Thus the model sees the fabricated knowledge as training text without memorizing the evaluation prompts.

The attribution split is separate from target-model exposure. Because the task is to attribute responses to training documents, target LLMs may see the full FAKEWIKI text corpus. We then split document identifiers 80/20 for training and evaluating attribution methods: SCORINGMODEL is trained on responses from training document IDs and evaluated on held-out document IDs. Source-preserving

variants are valid positives for the same document ID, while anti-documents are hard negatives and never valid attribution targets.

3.3 QA Probes and Target Responses

For each retained article, we generate five one-sentence questions whose answers are short phrases grounded in that article. After continued pretraining, we query each target LLM M with either the clean question q_D or a transformed version $J(q_D)$ and collect the response

$$r = M(J(q_D)).$$

An attribution method receives the response, optionally the original question, and the full candidate corpus. It must rank the documents that most likely support the response. Evaluation uses Recall@k, counting a prediction as correct if any valid source variant appears in the top-k.

3.4 Source Variants and Hard Negatives

Each source document is expanded into three variant families.

- **Paraphrases** preserve the original facts while changing wording and discourse structure. They are counted as valid positives during evaluation.
- **Retro-generated variants** are written from the QA facts rather than from the original article. They place answer-relevant information inside a different surrounding context, reducing simple lexical overlap with the source article.
- **Anti-documents** preserve the topic, entity name, style, and much of the wording of the original document, but delete or alter the facts needed to answer the QA probes. They are hard negatives: a method that only detects topical similarity should rank them highly, while a provenance method should not.

These variants separate documents that merely resemble the response from documents that actually support it.

3.5 Transformed Query Conditions

We evaluate five query conditions. The clean condition asks the original QA probe. The other four transformations stress different failure modes of provenance retrieval.

- `Obfuscate` replaces many content words with unrelated benign words while preserving the intended question through a mapping.
- `RolePlay` wraps the question inside a persona or scenario.
- `NoiseInjection` surrounds the question with unrelated filler text.
- `Indirect` rewrites the query into an indirect or multi-hop prompt with reduced surface overlap.

These transformations are controlled stress tests, not a complete jailbreak taxonomy. They ask whether attribution still works when prompt and response no longer expose clean lexical cues; full templates are in Appendix K.

4 Methods

4.1 Problem Setup

Let $\mathcal{D} = \{D_j\}_{j=1}^{N}$ be a candidate corpus that has been injected into a target LLM through continued pretraining, and let $P_i \subseteq \mathcal{D}$ denote the valid sources for example i, including the original document and valid variants. Given a question q_i, a transformed query $J(q_i)$, and a response r_i generated by the continued-pretrained target model, an attribution method produces a score $\phi(r_i, q_i, D_j)$ for each candidate document. We evaluate with

$$\text{Recall@}k = \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \mathbf{1}\left\{\text{Top}_k(\phi_i) \cap P_i \neq \emptyset\right\}. \tag{1}$$

All main results report Recall@10 as percentages.

4.2 SCORINGMODEL: Supervised Provenance Scoring

SCORINGMODEL is a supervised pairwise ranker that maps response-side and document-side features into a shared embedding space, rather than an N-way classifier over fixed document labels. For an input feature vector $\boldsymbol{x}$, a two-layer MLP f_θ produces a normalized embedding, and response-document compatibility is temperature-scaled cosine similarity:

$$s_\theta(r, D) = \frac{1}{\tau} \left\langle \frac{f_\theta(\boldsymbol{x}_r)}{\|f_\theta(\boldsymbol{x}_r)\|_2}, \frac{f_\theta(\boldsymbol{x}_D)}{\|f_\theta(\boldsymbol{x}_D)\|_2} \right\rangle. \tag{2}$$

We use target-LLM hidden states and QA-style text embeddings as input features, selecting the feature variant on validation data per target model. For each positive pair (r_i, D_i^+), the training batch contains three kinds of negatives. In-batch negatives provide cheap contrast against unrelated documents. Retrieval-mined hard negatives are topically or semantically close documents retrieved by an embedding model. Curated anti-documents are the most targeted negatives because they preserve surface similarity while removing answer support. Training minimizes an InfoNCE objective [Oord et al., 2018]:

$$\mathcal{L}_i = -\log \frac{\exp s_\theta(r_i, D_i^+)}{\exp s_\theta(r_i, D_i^+) + \sum_{D \in \mathcal{N}_i} \exp s_\theta(r_i, D)}. \tag{3}$$

At inference time, we precompute candidate document features and score each response by dot products in the learned space. Implementation details are in Appendix J.

4.3 Activation Steering with Retrieval Fusion

We also evaluate a training-free activation-based signal. Instead of asking only whether a response and document are textually similar, activation steering asks whether a candidate document points the target model's internal state toward the observed answer. Exact activation patching would require one intervention per candidate document, so STEERFUSE uses a cached-vector approximation.

Step 1: document activation directions. For a chosen layer $\ell^\star$, each candidate document D induces an activation direction by mean-pooling the hidden states produced when the target model reads the document:

$$\boldsymbol{v}_D = \frac{\sum_{t \in D} a_t \boldsymbol{h}_t^{(\ell^\star)}}{\sum_{t \in D} a_t}, \tag{4}$$

where a_t is the attention mask. Document directions are normalized and cached before scoring.

Step 2: response-side proxy and activation score. For a response $r = y_{1:m}$, we use an efficient response-side proxy given by the sum of LM-head rows for the generated answer tokens:

$$\tilde{\boldsymbol{g}}_r = \sum_{i=1}^{m} \boldsymbol{W}_{y_i}. \tag{5}$$

Each candidate document is scored by cosine similarity between this response-side proxy and the cached document direction:

$$s_{\text{act}}(r, D) = \cos(\tilde{\boldsymbol{g}}_r, \boldsymbol{v}_D). \tag{6}$$

This reduces retrieval to cosine similarity against cached document vectors, avoiding patched forward or backward passes.

Step 3: approximate intervention score. The activation score approximates a direct intervention question: would steering the answer-token hidden states toward $\boldsymbol{v}_D$ increase the likelihood of the observed answer? If we patch

$$\tilde{\boldsymbol{h}}_{t_i}^{(D)}(\alpha) = (1 - \alpha)\boldsymbol{h}_{t_i}^{(\ell^\star)} + \alpha \boldsymbol{v}_D, \tag{7}$$

then a first-order expansion of the answer log-probability gain gives

$$\Delta_{\text{exact}}(D) \approx \alpha \sum_{i=1}^{m} \boldsymbol{g}_i^\top (\boldsymbol{v}_D - \boldsymbol{h}_{t_i}^{(\ell^\star)}) \propto \boldsymbol{g}^\top \boldsymbol{v}_D, \qquad \boldsymbol{g} = \sum_{i=1}^{m} \boldsymbol{g}_i. \tag{8}$$

Thus document ranking can be approximated by dot products with cached document directions. When $\ell^\star$ is the final layer before the LM head, the token sensitivity has the form

$$\boldsymbol{g}_i = \boldsymbol{W}_{y_i} - \mathbb{E}_{w \sim p_\theta(\cdot|y_{<i})}[\boldsymbol{W}_w] \approx \boldsymbol{W}_{y_i}, \tag{9}$$

which motivates the implemented proxy $\tilde{\boldsymbol{g}}_r = \sum_i \boldsymbol{W}_{y_i}$. Appendix D gives the full derivation and discusses the approximation.

Step 4: retrieval fusion. Text retrieval is a strong prior but can reward resemblance rather than source support. We therefore fuse SBERT-QA similarity with s_{act}, using validation-tuned z-score fusion or reciprocal-rank fusion for STEERFUSE. The main comparison gives STEERFUSE this inference-time fusion, while SCORINGMODEL is reported without inference-time fusion.

5 Experimental Setup

Target models. We evaluate nine open-weight instruction-tuned LLMs: `TinyLlama-1.1B-Chat-v1.0`, `Llama-3.2-1B-Instruct`, `Qwen2-1.5B-Instruct`, `Llama-3.2-3B-Instruct`, `Qwen2.5-7B-Instruct`, `Llama-2-7b-chat-hf`, `Mistral-7B-Instruct-v0.3`, `Llama-3.1-8B-Instruct`, and `Qwen3-8B`.

Target-model continued pretraining. Before collecting responses, each target LLM is continued-pretrained on the FAKEWIKI text corpus with a causal language-modeling objective for 3 epochs at learning rate 2×10^{-5}. The training text consists of fabricated article text and document variants, not the QA probes or transformed queries. This exposure step is what makes the downstream task training-data attribution: the model responses are elicited from LLMs that have encountered the candidate documents as training text. The 80/20 document-ID split used later is for training and evaluating attribution methods, not for withholding documents from the target LLM.

Baselines. We compare against eleven retrieval baselines. MinHash estimates lexical resemblance by hashing token shingles and comparing compact signatures, giving a fast approximation to Jaccard overlap [Broder, 1997]; we run it over the answer alone and over the question-answer pair. The finetuned dense LLM embedding baseline uses an in-domain embedding head and ranks documents by cosine similarity. The remaining dense retrieval baselines use SBERT [Reimers and Gurevych, 2019], BGE [Xiao et al., 2024], and Contriever [Izacard et al., 2022]: MiniLM, MPNet, BGE-base, and Contriever are evaluated with answer-only queries and with QA queries. Answer-only retrieval tests whether the generated response itself carries enough source evidence; QA retrieval tests a stronger setting where the retriever also sees the original question. In aggregate tables, "best baseline" means the strongest of these eleven baselines for that model and query condition.

Metrics and selection. The main paper reports Recall@10, with full per-model R@1 and R@5 tables in Appendix I. For SCORINGMODEL, feature mode and checkpoint selection use held-out Clean validation data per target model, and the selected no-fusion scorer is evaluated unchanged on transformed queries. For STEERFUSE, validation selects the retrieval-fusion setting per target model and query condition. Further implementation details are in Appendix J.

6 Main Results

6.1 Aggregate Recall@10 Results

Table 2 summarizes Recall@10 averaged across all nine target models. SCORINGMODEL improves over the strongest baseline in every query condition. The gains are largest for `RolePlay` and `NoiseInjection`, where surface retrieval remains plausible but unreliable, and smallest for `Obfuscate` and `Indirect`, which represent two different hard cases: shallow lexical substitution can sometimes favor response-only semantic retrieval, while indirect prompting reduces all methods to low absolute recall.

Across all 9×5 cells, SCORINGMODEL beats the best baseline in 41/45 cells and beats STEERFUSE in 40/45 cells. Mean transformed-query Recall@10 improves by 13.2 points over the best baseline. Additional aggregate summaries, including full win counts and transformed-query averages by target model, are reported in Appendix E.

Table 2: Recall@10 averaged across nine target LLMs. The best baseline is selected from eleven retrieval baselines separately for each model and query condition. Best per row is in **bold**, second-best is underlined.

Query condition	Best baseline	STEERFUSE	SCORINGMODEL	Δ vs. baseline	Δ vs. STEERFUSE
Clean	55.7	69.2	**77.2**	+21.5	+8.0
Obfuscate	39.1	30.5	**44.4**	+5.3	+13.9
RolePlay	42.9	50.1	**62.5**	+19.6	+12.4
NoiseInjection	36.7	47.0	**59.2**	+22.5	+12.2
Indirect	12.0	14.5	**17.7**	+5.7	+3.2
Average	37.3	42.3	**52.2**	+14.9	+9.9

Table 3: Large-model Recall@10 table. Rows include all baselines and our attribution methods for two instruction-tuned target models. Best per column is in **bold**, second-best is underlined.

Method	Qwen3-8B					Llama-3.1-8B				
	Clean	Obfuscate	RolePlay	NoiseInjection	Indirect	Clean	Obfuscate	RolePlay	NoiseInjection	Indirect
MinHash (answer)	0.1	0.1	0.1	0.1	0.1	0.0	0.1	0.1	0.1	0.0
MinHash (QA)	0.1	0.1	0.1	0.1	0.1	0.0	0.1	0.1	0.1	0.0
Finetuned EmbedSim	17.4	9.4	16.1	16.4	2.3	16.9	15.9	23.6	23.1	2.2
SBERT-MiniLM (answer)	28.4	40.6	23.5	23.9	9.3	3.7	35.7	10.6	12.9	3.5
SBERT-MPNet (answer)	25.5	36.3	20.3	20.1	8.0	3.6	32.4	9.7	11.6	3.0
SBERT-MiniLM (QA)	39.7	12.4	29.1	25.0	9.1	28.9	9.5	19.9	18.5	4.0
SBERT-MPNet (QA)	38.1	3.7	27.5	14.8	7.9	29.1	3.9	21.6	9.8	3.5
BGE-base (answer)	30.8	43.0	26.0	25.7	11.4	4.3	37.3	11.7	13.1	3.8
BGE-base (QA)	43.1	16.6	40.3	23.6	11.3	33.3	13.4	30.6	13.8	5.9
Contriever (answer)	32.4	42.8	26.2	26.6	10.6	3.6	27.9	11.3	12.5	2.7
Contriever (QA)	30.1	22.8	22.3	15.0	3.5	5.1	10.8	9.3	7.1	0.8
STEERFUSE	76.4	30.1	60.5	51.1	20.1	57.8	37.7	43.5	38.1	9.6
SCORINGMODEL	**78.1** ±0.5	**53.9** ±1.5	**63.8** ±0.7	**62.1** ±0.3	**21.6** ±0.8	**76.7** ±0.2	**59.5** ±2.1	**63.4** ±0.6	**63.5** ±0.4	**18.2** ±0.4

Seed robustness. We re-train each Clean-validation-selected SCORINGMODEL configuration with three seeds. The average per-cell standard deviation is 0.94 Recall@10 points, with variance largest on `Obfuscate`; full mean±std results are in Appendix G.

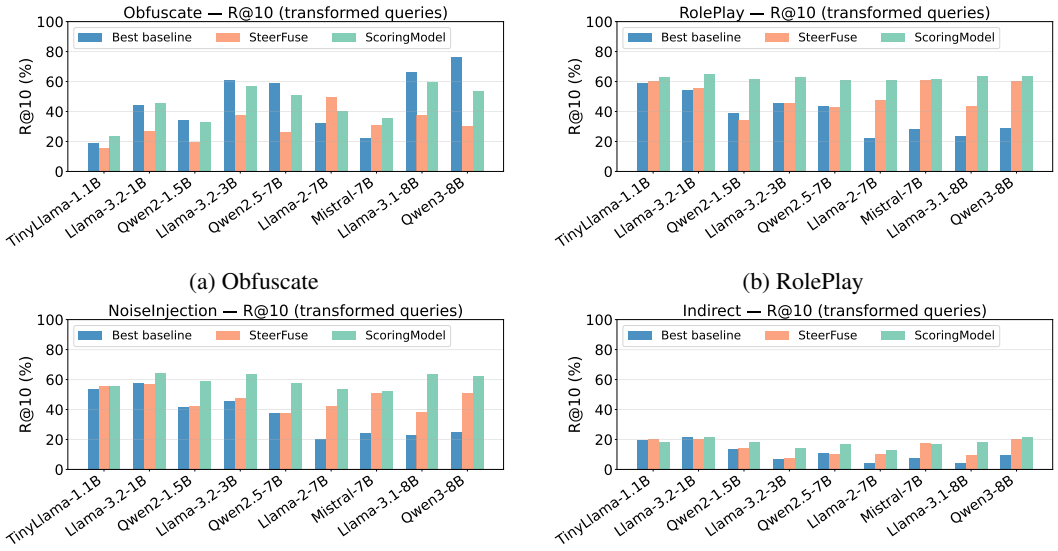
6.2 Per-Method Results on Large Models

Table 3 reports Recall@10 for `Qwen3-8B` and `Llama-3.1-8B`, with methods as rows and query conditions as columns. These two models are useful stress cases because the best retrieval baseline varies across conditions: QA-style dense retrieval is strongest on clean prompts, while answer-only BGE or SBERT can become competitive under `Obfuscate`. Even against this condition-specific best-of-baselines comparison, SCORINGMODEL wins all ten large-model columns, while STEERFUSE is the strongest non-supervised method in most non-obfuscation columns. The remaining seven target-model tables are reported in Appendix H, and the aggregate trends across all models are summarized above.

The table shows that clean retrieval can be deceptively strong, but transformed prompts expose baseline instability: methods that work under clean prompting may fall sharply under indirect prompting or noise injection, and obfuscation can flip the strongest baseline from QA retrieval to answer-only retrieval. By contrast, SCORINGMODEL remains high across all five query conditions for both large models, including settings where the best baseline is below 15 Recall@10. The corresponding Recall@1 and Recall@5 tables in Appendix I show that this advantage is not only a top-10 effect: SCORINGMODEL also wins every stricter-cutoff column for `Qwen3-8B` and `Llama-3.1-8B`.

6.3 Scaling on Transformed Prompts

Figure 2 visualizes the per-model improvement of SCORINGMODEL over the best baseline on the four transformed query conditions. The largest gains appear for the larger target models: `Llama-3.1-8B` improves by +26.9 Recall@10 on average over transformed conditions, `Qwen3-8B` by +20.0, `Llama-2-7B` by +19.0, and `Mistral-7B` by +17.0. Smaller models still benefit, but with narrower margins. This pattern suggests that larger LLM hidden states encode more recoverable provenance information, making attribution easier for a trained scorer even as the underlying model is larger.



(a) Obfuscate

(b) RolePlay

(c) NoiseInjection

(d) Indirect

Figure 2: Improvement of SCORINGMODEL over the best baseline on transformed query conditions.

6.4 Ablation Study

Appendix F provides full ablations. The main takeaway is that no-fusion SCORINGMODEL is already a strong standalone scorer: it uses a single learned compatibility score, requires no test-time mixing weight, and wins 41/45 model-by-condition cells. We evaluate SCORINGMODEL–SBERT fusion only as an ablation.

STEERFUSE shows a different pattern: its mean gain over the stronger of activation-only and SBERT-only rankings is only +2.0 Recall@10, and about 96% of the fusion uplift comes from SBERT. The main exception is `Obfuscate`, where the activation component contributes about +7.5 points, suggesting that internal-state evidence helps most when lexical cues are actively distorted.

Other ablations show that the conclusions are not driven by a single representation or combiner: z-score and reciprocal-rank fusion give similar trends, and feature-mode diagnostics show complementary hidden-state and QA-style signals.

7 Conclusion

We presented DATADIGNITY, a benchmark and method suite for robust pinpoint provenance in LLM responses. FAKEWIKI shows that clean retrieval can overstate attribution reliability: under obfuscation, role-play, noise, and indirect prompts, methods must distinguish true answer support from topical resemblance. This distinction matters for audit workflows because a high-similarity document is not necessarily the document that supplied the answer-critical fact. In this setting, SCORINGMODEL improves mean Recall@10 from 37.3 to 52.2 and wins 41/45 model-by-condition cells, with the same trend holding at stricter Recall@1 and Recall@5 cutoffs. The gains are largest on transformed prompts and larger target models, suggesting that supervised attribution can recover provenance signals that generic retrievers miss when surface cues are unstable. STEERFUSE shows that activation-space evidence is useful when stabilized by text retrieval, but is less uniform than the supervised scorer; this points to internal-state evidence as a promising complement rather than a complete replacement for retrieval.

Overall, robust training data attribution should be evaluated with hard negatives and prompt transformations that expose failures hidden by clean semantic retrieval. We view pinpoint provenance systems as audit aids that return inspectable candidate sources, and future work should push beyond document-level recall toward calibrated confidence and finer-grained evidence localization.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Ekin Akyürek, Tolga Bolukbasi, Frederick Liu, Binbin Xiong, Ian Tenney, Jacob Andreas, and Kelvin Guu. Towards tracing factual knowledge in language models back to the training data. *arXiv preprint arXiv:2205.11482*, 2022.

Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. Relatif: Identifying explanatory training samples via relative influence. In *International Conference on Artificial Intelligence and Statistics*, pages 1899–1909. PMLR, 2020.

Andrei Z. Broder. On the resemblance and containment of documents. In *Proceedings of Compression and Complexity of Sequences*, pages 21–29. IEEE, 1997.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023.

Milad Fotouhi, Mohammad Taha Bahadori, Oluwaseyi Feyisetan, Payman Arabshahi, and David Heckerman. Fast training dataset attribution via in-context learning. *arXiv preprint arXiv:2408.11852*, 2024.

Xiaochuang Han and Yulia Tsvetkov. Influence tuning: Demoting spurious correlations via instance attribution and instance-driven updates. *arXiv preprint arXiv:2110.03212*, 2021.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

Muhammad Khalifa, David Wadden, Emma Strubell, Honglak Lee, Lu Wang, Iz Beltagy, and Hao Peng. Source-aware training enables knowledge attribution in language models. *arXiv preprint arXiv:2404.01019*, 2024.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. *arXiv preprint arXiv:2310.00902*, 2023.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, 2023.

Xiaomin Li, Mingye Gao, Zhiwei Zhang, Chang Yue, and Hong Hu. Selection of llm fine-tuning data based on orthogonal rules. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 31760–31768, 2026.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. In *International Conference on Machine Learning*, pages 27074–27113. PMLR, 2023.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930, 2020.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*, 2019.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992, 2019.

Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. Extracting latent steering vectors from pretrained language models. *arXiv preprint arXiv:2205.05124*, 2022.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Cheng Wang, Xinyang Lu, See-Kiong Ng, and Bryan Kian Hsiang Low. Trace: Transformer-based attribution using contrastive embeddings in llms, 2024. URL `https://arxiv.org/abs/2407.04981`.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649, 2024.

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. Catastrophic failure of llm unlearning via quantization. *arXiv preprint arXiv:2410.16454*, 2024.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Limitations and Future Work

Limitations. Our study provides a controlled and fine-grained evaluation of pinpoint provenance, with ground-truth source documents, hard negatives, source-preserving variants, and transformed query conditions that stress different attribution failure modes. However, several aspects could be further improved. First, our transformed query conditions are controlled stress tests rather than a full taxonomy of user prompting behavior. They are designed to isolate different shortcut pressures, not to exhaust every possible way a prompt can alter the surface form of a response. Second, our main metric is Recall@10, which is appropriate for audit workflows where a human inspector reviews a short candidate list, but it does not capture all downstream notions of attribution usefulness, such as calibrated confidence, explanation quality, or evidence localization within a document. Finally, our evaluation focuses on document-level attribution. This is useful for source inspection, but it does not identify the exact sentence or span that supports a response.

Future work. Future work can address these limitations in three directions. First, the transformed-query suite can be expanded to cover a broader range of prompt styles, including more natural user rewrites, multi-turn interactions, and compositional transformations that combine obfuscation, role-play, noise, and indirect prompting. Second, evaluation can move beyond Recall@10 by adding calibration and ranking-quality metrics, so that provenance systems report not only whether a true source appears in the top candidates, but also how reliable the returned evidence is. Third, future benchmarks can extend document-level attribution to finer-grained evidence localization, such as sentence-level or span-level provenance, which would make the returned results easier for human auditors to verify.

B Broader Impacts

This work is intended to support more reliable auditing of language-model outputs by helping identify candidate source documents that may support a generated response. Potential positive impacts include improved dataset curation, copyright and provenance analysis, misinformation forensics, and safety debugging. At the same time, provenance scores should be interpreted carefully: a high-ranked document is evidence of source support within the candidate corpus, not a definitive causal claim about model training or generation. Misinterpreting attribution results could lead to overconfident conclusions about whether a particular document caused an output. We therefore frame DATADIGNITY as an audit aid that returns inspectable evidence for human review, rather than as an automated system for assigning legal or causal responsibility.

C Asset Licenses

We use publicly available models and software tools, including the open-weight target LLMs listed in Section 5, Sentence-BERT embedding models [Reimers and Gurevych, 2019], BGE embeddings [Xiao et al., 2024], Contriever embeddings [Izacard et al., 2022], PyTorch, CUDA, and standard Python libraries. We cite the corresponding model families, methods, and software tools where appropriate, and use these assets for research evaluation under their respective licenses and terms of use. The newly introduced FAKEWIKI benchmark consists of fabricated documents generated for this study, and the anonymized release includes the benchmark data, prompts, and evaluation code.

D Activation-Steering Approximation Details

This appendix expands the intervention view of STEERFUSE from Section 4.3. The method starts from an exact but expensive activation-patching score, then uses a sequence of approximations: a first-order Taylor approximation that turns patched forward passes into dot products, and an LM-head approximation that avoids a backward pass for the sensitivity vector by replacing the gradient direction with answer-token LM-head rows.

D.1 Exact Activation-Patching Score

Fix a target response $y_{1:m}$ and a candidate document direction $\boldsymbol{v}_D$. At the selected layer $\ell^\star$, let $\boldsymbol{h}_{t_i}^{(\ell^\star)}$ be the hidden state used to predict answer token y_i. The exact intervention asks whether mixing this candidate direction into the answer-token hidden states increases the model probability of the observed response:

$$\tilde{\boldsymbol{h}}_{t_i}^{(D)}(\alpha) = (1-\alpha)\boldsymbol{h}_{t_i}^{(\ell^\star)} + \alpha \boldsymbol{v}_D, \qquad \alpha \in (0,1]. \tag{10}$$

The corresponding exact score is

$$\Delta_{\text{exact}}(D) = \sum_{i=1}^{m} \left[\log p_\theta(y_i \mid y_{<i}; \tilde{\boldsymbol{h}}_{t_i}^{(D)}(\alpha)) - \log p_\theta(y_i \mid y_{<i}; \boldsymbol{h}_{t_i}^{(\ell^\star)}) \right]. \tag{11}$$

A large positive value means that injecting document D makes the target model more confident in the fixed response. This is the most direct causal-style score, but evaluating it for N documents requires one baseline forward pass plus N patched forward passes.

D.2 First-Order Taylor Approximation

To avoid a patched forward pass for every candidate document, define the scalar objective

$$f(\{\boldsymbol{h}_{t_i}\}_{i=1}^m) = \sum_{i=1}^{m} \log p_\theta(y_i \mid y_{<i}), \tag{12}$$

the log-probability of the whole fixed answer under the unpatched hidden states. A first-order Taylor expansion around the original hidden states gives

$$f(\tilde{\boldsymbol{h}}^{(D)}) - f(\boldsymbol{h}) = \alpha \sum_{i=1}^{m} \underbrace{\nabla_{\boldsymbol{h}_{t_i}^{(\ell^\star)}} f(\boldsymbol{h})^\top}_{\boldsymbol{g}_i^\top} (\boldsymbol{v}_D - \boldsymbol{h}_{t_i}^{(\ell^\star)}) + O(\alpha^2) \tag{13}$$

$$= \alpha \left(\sum_{i=1}^{m} \boldsymbol{g}_i \right)^\top \boldsymbol{v}_D - \alpha \sum_{i=1}^{m} \boldsymbol{g}_i^\top \boldsymbol{h}_{t_i}^{(\ell^\star)} + O(\alpha^2). \tag{14}$$

The second term does not depend on D, so it does not affect the ranking of candidate documents. If

$$\boldsymbol{g} = \sum_{i=1}^{m} \boldsymbol{g}_i = \sum_{i=1}^{m} \frac{\partial}{\partial \boldsymbol{h}_{t_i}^{(\ell^\star)}} \log p_\theta(y_i \mid y_{<i}), \tag{15}$$

then the document-dependent part of the first-order score is

$$\widehat{\Delta \log p}(D) \propto \boldsymbol{g}^\top \boldsymbol{v}_D. \tag{16}$$

This approximation turns document ranking into a matrix-vector product once all document vectors have been cached. It is accurate when the mixing weight is small enough that higher-order terms in α do not dominate the ranking.

D.3 Approximating Sensitivity Without Backpropagation

Computing $\boldsymbol{g}$ exactly requires one backward pass for each response. This is much cheaper than N patched forward passes, but it can still be expensive when many responses must be attributed. When the patched layer is the final hidden layer before the language-model head, the per-token sensitivity can be written in closed form. Let $\boldsymbol{W} \in \mathbb{R}^{|V| \times d}$ be the LM-head matrix and let $\boldsymbol{W}_w$ denote the row associated with vocabulary token w. For the hidden state $\boldsymbol{h}$ at an answer position, the logits are

$$z_w = \boldsymbol{h}^\top \boldsymbol{W}_w, \qquad p_\theta(w \mid \mathrm{ctx}) = \frac{\exp(\boldsymbol{h}^\top \boldsymbol{W}_w)}{\sum_{u \in V} \exp(\boldsymbol{h}^\top \boldsymbol{W}_u)}. \tag{17}$$

Therefore, for answer token y_i,

$$\boldsymbol{g}_i = \frac{\partial}{\partial \boldsymbol{h}} \log p_\theta(y_i \mid y_{<i}) \tag{18}$$

$$= \boldsymbol{W}_{y_i} - \sum_{w \in V} p_\theta(w \mid y_{<i}) \boldsymbol{W}_w \tag{19}$$

$$= \boldsymbol{W}_{y_i} - \mathbb{E}_{w \sim p_\theta(\cdot \mid y_{<i})}[\boldsymbol{W}_w]. \tag{20}$$

This formula shows that the sensitivity direction points toward the LM-head row of the observed answer token and away from the probability-weighted average LM-head direction. Computing the expectation exactly requires multiplying the full vocabulary probability vector by $\boldsymbol{W}$, which can be costly. A further approximation drops this expectation term and uses

$$\tilde{\boldsymbol{g}}_i \propto \boldsymbol{W}_{y_i}, \qquad \tilde{\boldsymbol{g}} = \sum_{i=1}^{m} \tilde{\boldsymbol{g}}_i. \tag{21}$$

This is reasonable when the answer token direction dominates the local gradient, but it is not an exact identity: the omitted expectation term can matter when the next-token distribution is diffuse or when competing tokens have substantial probability mass. Under this approximation, the Taylor score becomes

$$\widehat{\Delta \log p}(D) \propto \tilde{\boldsymbol{g}}^\top \boldsymbol{v}_D, \tag{22}$$

so ranking still reduces to a dot product, but the response-side vector can be obtained by LM-head row lookups rather than backpropagation.

D.4 Implemented LM-Head Row Proxy

The gradient score in Eq. (16) removes the need for one patched forward pass per document, and the LM-head approximation in Eq. (21) removes the backward pass. The implemented STEERFUSE method represents the answer by the normalized sum of its generated-token LM-head rows and ranks documents by cosine similarity against cached document directions:

$$s_{\mathrm{act}}(r, D) = \cos\left(\sum_{i=1}^{m} \boldsymbol{W}_{y_i}, \boldsymbol{v}_D\right). \tag{23}$$

In experiments, we use a finer-grained version of the same score: long documents are split into sentence-respecting chunks, each chunk receives a score, and the document score is the maximum over its chunks. This proxy is cheaper than exact patching and avoids backpropagation, but it is still noisy because LM-head rows are only an approximation to the full gradient sensitivity in Eq. (20). For this reason, the main STEERFUSE method fuses the activation score with SBERT-QA retrieval rather than using the activation score alone.

E Additional Aggregate Results

Table 4 reports head-to-head win counts over all 45 model-by-query-condition cells. The pattern is consistent with the main aggregate results: SCORINGMODEL is not only better on average, but also wins across most individual settings. It outperforms the strongest retrieval baseline in 41/45 cells and outperforms STEERFUSE in 40/45 cells. At the same time, STEERFUSE beats the best baseline in 32/45 cells, suggesting that activation-based evidence can be useful when stabilized by retrieval fusion.

Table 4: Head-to-head win counts over all 45 model-by-query-condition cells.

Comparison	Wins / 45	Percent
SCORINGMODEL > best baseline	41	91%
SCORINGMODEL > STEERFUSE	40	89%
STEERFUSE > best baseline	32	71%

Table 5 averages Recall@10 across the four transformed query conditions, excluding clean prompts. This table highlights where the robustness gains are largest. The biggest improvements occur for Llama-3.1-8B and Qwen3-8B, where SCORINGMODEL improves over the best baseline by +26.9 and +20.0 points respectively. The next strongest gains appear for Llama-2-7B and Mistral-7B. Smaller models still benefit, but with narrower margins. This supports the main-paper observation that larger target models tend to expose more recoverable provenance signal for a supervised scorer, especially when surface-form cues are disrupted.

Table 5: Average transformed-query Recall@10 by target model. Clean prompts are excluded.

Model	Best baseline	STEERFUSE	SCORINGMODEL	Δ vs. baseline
Llama-3.1-8B	24.2	32.2	**51.1**	+26.9
Qwen3-8B	30.3	40.4	**50.4**	+20.0
Llama-2-7B	23.0	37.3	**42.0**	+19.0
Mistral-7B	24.6	40.2	**41.5**	+17.0
Qwen2-1.5B	32.1	27.6	**43.0**	+10.9
Llama-3.2-3B	39.9	34.6	**49.4**	+9.5
Qwen2.5-7B	37.9	29.4	**46.6**	+8.8
Llama-3.2-1B	44.4	40.0	**49.2**	+4.8
TinyLlama-1.1B	37.9	38.2	**40.3**	+2.4

Overall, these aggregate results show that the gains are not driven by a small number of favorable cases. SCORINGMODEL is consistently strong across both head-to-head comparisons and transformed-query averages, while STEERFUSE provides a useful but less reliable intermediate signal.

F Ablation Details

Inference-time fusion. Adding SBERT fusion to SCORINGMODEL improves mean Recall@10 by +5.5 points over the stronger of no-fusion SCORINGMODEL and SBERT alone. The gains are distributed across query conditions: Clean +2.7, Obfuscate +5.5, RolePlay +9.5, NoiseInjection +8.6, and Indirect +1.0. We keep this variant as an ablation because the no-fusion method is simpler, requires no test-time fusion weight, and already wins the main comparison.

STEERFUSE decomposition. For STEERFUSE, the mean fusion gain over the stronger of activation-only and SBERT-only rankings is +2.0 Recall@10, and approximately 96% of the fusion uplift comes from the SBERT component. The main exception is `Obfuscate`, where the activation component contributes about +7.5 points. Thus activation steering is informative, but not yet a strong standalone provenance ranker.

Combiner choice. Z-score fusion wins the majority of SCORINGMODEL-fusion cells and STEERFUSE cells, so the main paper uses z-score when reporting STEERFUSE. Reciprocal-rank fusion is reported as a sensitivity check and is most useful under `Indirect`, where score distributions shift strongly.

Feature mode. We consider three input-feature variants for the SCORINGMODEL scorer. The `llm` feature is a mean-pooled hidden state from a selected layer of the target LLM, computed for the response side and for the candidate document side; it is intended to capture model-internal content representations that may survive obfuscation or paraphrase. The `qa` feature is a SBERT-MiniLM embedding [Reimers and Gurevych, 2019] of a QA-style textual representation, which is strong when the question–answer pair retains semantic alignment with the source. The `concat` feature concatenates the two, allowing early fusion at the MLP input. We emphasize that all three variants are evaluated at $\lambda = 0$: the MLP produces a single learned score per candidate, and there is no test-time score-level mixing with a separate SBERT ranker. Main results use the feature mode selected on Clean validation for each target model. The condition-wise feature results in Figure 3 are diagnostic rather than a transformed-condition tuning procedure.

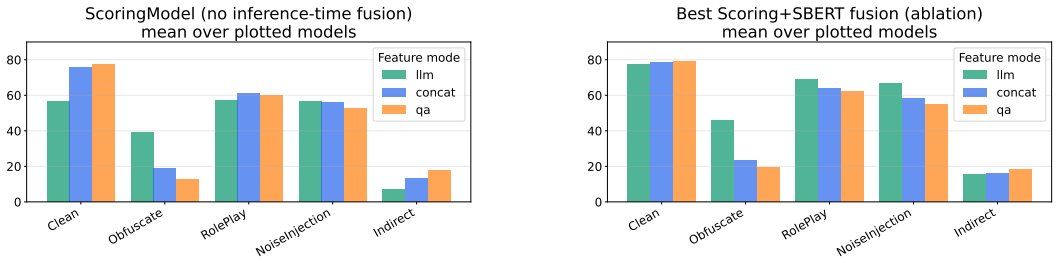
Grid size and embedding backbone. A wider negative-mining grid is sufficient for the `concat` setting; larger grids help `llm` features, especially on `Obfuscate`. MiniLM is stronger than MPNet on this benchmark across the relevant sweeps.

The figures below visualize the same ablation questions: which input representation is most useful for SCORINGMODEL, whether z-score fusion or reciprocal-rank fusion is the more important design choice, and how much the validation-selected fusion models rely on the learned attribution signal versus SBERT. All plots report Recall@10 and average over the nine target models unless otherwise stated.

Feature representation. Figure 3 compares three feature modes for SCORINGMODEL. The `qa` feature is strongest on clean prompts in the no-fusion setting, reaching roughly the high-70s Recall@10, because the question-answer text remains semantically aligned with the source document. The same text channel is also the best no-fusion feature on `Indirect`, where all methods are low but QA semantics still preserve the most usable retrieval signal. By contrast, the `llm` feature is clearly strongest on `Obfuscate`: its average Recall@10 is roughly twice that of `concat` and more than three times that of `qa`. This supports the interpretation that target-model hidden states carry provenance information that is less tied to surface word choice. For `RolePlay` and `NoiseInjection`, the differences are smaller: `concat` is best on role-play without fusion, while `llm` and `concat` are close on noise injection. This is consistent with these transformations preserving more of the original semantics while adding stylistic or irrelevant context.

The fusion ablation in Figure 3 changes the picture in an informative way. Once SBERT score-level fusion is allowed, `llm` becomes the best feature mode for `Obfuscate`, `RolePlay`, and `NoiseInjection`. The text-only `qa` mode remains competitive on clean prompts and remains best on `Indirect`, but it no longer dominates the transformed settings. Thus the best fusion behavior does not come from replacing hidden-state features with text retrieval; it comes from using SBERT as a stabilizing semantic prior while letting model-internal features contribute complementary signal

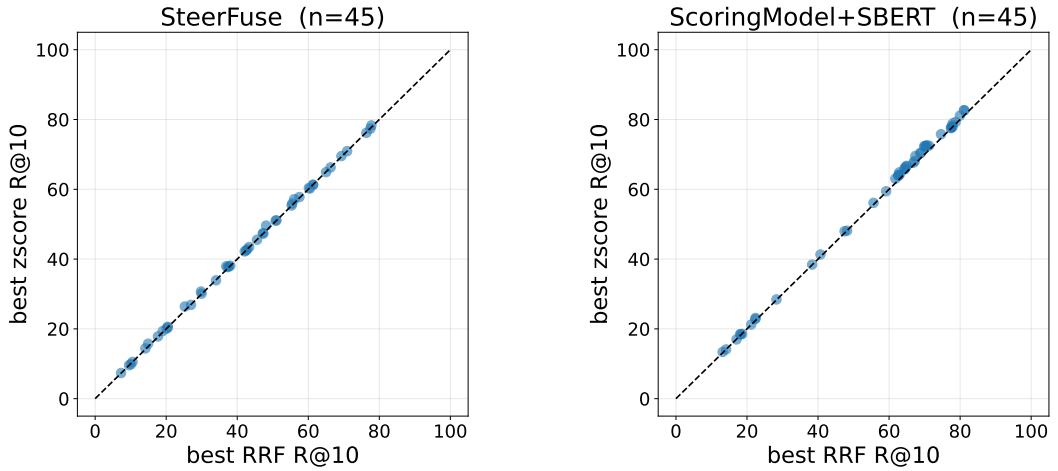
under transformed prompts. This is why the main paper reports the simpler no-fusion SCORINGMODEL as the primary method and treats SCORINGMODEL +SBERT as an ablation: fusion can improve robustness, but the learned scorer already provides a strong standalone attribution signal.



(a) No-fusion SCORINGMODEL (b) Fusion ablation

Figure 3: Feature-mode ablations for SCORINGMODEL. Left: no inference-time fusion. Right: best SCORINGMODEL +SBERT fusion. Hidden-state features are most useful under obfuscation and, with fusion, become the strongest feature mode on most transformed conditions.

Fusion combiner. Figure 4 compares the two score-combination rules used in the fusion sweeps: z-score fusion and reciprocal-rank fusion (RRF). Each point is one model-by-query-condition cell, so each panel contains 45 points. In both the STEERFUSE panel and the SCORINGMODEL +SBERT panel, almost all points lie on or very near the diagonal. This means that the choice of combiner is not the main driver of performance: if a cell is easy or hard for fusion, it is usually easy or hard under both normalization schemes. The practical implication is that the results are not an artifact of a fragile score calibration trick. Z-score fusion is slightly more favorable in some high-recall SCORINGMODEL cells, while RRF is occasionally competitive when raw score scales are less comparable, but the two views tell the same qualitative story.

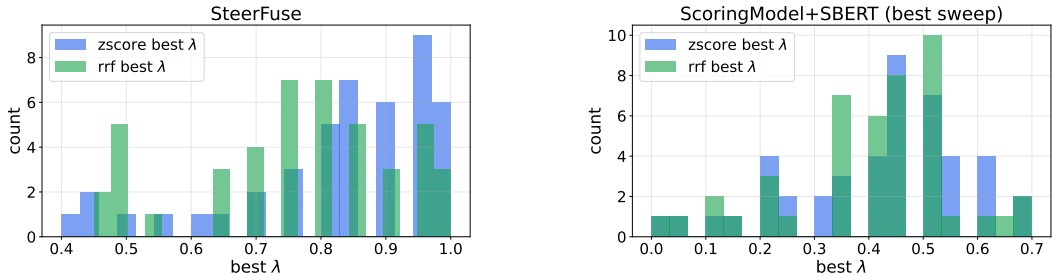


(a) STEERFUSE (b) SCORINGMODEL fusion ablation

Figure 4: Z-score versus reciprocal-rank fusion over all 45 model-by-condition cells. Points near the diagonal indicate that the fusion gains are not sensitive to the particular combiner.

Fusion weights. Figure 5 shows the validation-selected mixing weights. Here $\lambda = 0$ means the method-only score is used, $\lambda = 1$ means SBERT-QA alone is used, and intermediate values indicate genuine score-level fusion. The STEERFUSE histogram is concentrated toward large λ values, mostly around 0.75–1.0, with no mass near small λ. This confirms that raw activation steering is a useful but noisy signal: validation usually prefers to keep a large SBERT component and add steering only as a correction. The SCORINGMODEL +SBERT histogram is different. Its selected weights are mostly intermediate, with substantial mass around 0.35–0.55 and little mass near $\lambda = 1$. This indicates

that the learned SCORINGMODEL signal remains central even when SBERT fusion is permitted; validation rarely chooses to discard the learned scorer in favor of SBERT alone. The contrast between the two histograms is therefore important: STEERFUSE fusion is largely SBERT-anchored, whereas SCORINGMODEL fusion is closer to true complementarity between a learned attribution scorer and a semantic retrieval prior.



(a) STEERFUSE (b) SCORINGMODEL fusion ablation

Figure 5: Validation-selected fusion weights. $\lambda = 0$ is method-only and $\lambda = 1$ is SBERT-only. STEERFUSE selects large λ values, while SCORINGMODEL +SBERT selects intermediate values, indicating more genuine complementarity.

G Seed Robustness

We report seed robustness for SCORINGMODEL because it is the only learned method with stochastic training; STEERFUSE is deterministic given the target-model forward passes, cached activations, and validation-selected fusion setting. Stability of STEERFUSE is instead assessed through the fusion-combiner and fusion-weight ablations in Appendix F.

For each target model we re-train the Clean-validation-selected SCORINGMODEL configuration under three independent seeds (42, 123, 2024) and report Recall@10 mean with standard deviation shown as a subscript per (model, query condition). Standard deviations are typically below 1.5 Recall@10 points; the larger values on `Obfuscate` reflect higher sensitivity of the synonym-substitution condition to negative-mining randomness.

Table 6: SCORINGMODEL Recall@10 mean with standard deviation shown as a subscript across three seeds per (model, query condition).

Model	Clean	Obfuscate	RolePlay	NoiseInjection	Indirect
TinyLlama-1.1B	$78.2_{\pm 0.2}$	$23.8_{\pm 1.2}$	$62.7_{\pm 1.1}$	$55.9_{\pm 1.0}$	$18.6_{\pm 0.3}$
Llama-3.2-1B	$76.5_{\pm 0.5}$	$45.8_{\pm 2.6}$	$64.7_{\pm 0.3}$	$64.5_{\pm 1.0}$	$21.8_{\pm 0.7}$
Qwen2-1.5B	$77.2_{\pm 0.4}$	$33.2_{\pm 4.4}$	$61.4_{\pm 1.4}$	$59.1_{\pm 1.0}$	$18.4_{\pm 0.9}$
Llama-3.2-3B	$76.5_{\pm 0.5}$	$56.8_{\pm 4.3}$	$62.8_{\pm 0.1}$	$63.9_{\pm 0.5}$	$14.1_{\pm 0.3}$
Qwen2.5-7B	$77.2_{\pm 0.5}$	$50.8_{\pm 1.2}$	$61.0_{\pm 0.8}$	$57.9_{\pm 0.2}$	$16.9_{\pm 1.6}$
Llama-2-7B	$76.9_{\pm 0.4}$	$40.4_{\pm 0.7}$	$61.0_{\pm 0.9}$	$53.9_{\pm 1.1}$	$12.7_{\pm 0.4}$
Mistral-7B	$77.7_{\pm 0.4}$	$35.5_{\pm 1.3}$	$61.6_{\pm 0.6}$	$52.1_{\pm 1.6}$	$16.9_{\pm 0.5}$
Llama-3.1-8B	$76.7_{\pm 0.2}$	$59.5_{\pm 2.1}$	$63.4_{\pm 0.6}$	$63.5_{\pm 0.4}$	$18.2_{\pm 0.4}$
Qwen3-8B	$78.1_{\pm 0.5}$	$53.9_{\pm 1.5}$	$63.8_{\pm 0.7}$	$62.1_{\pm 0.3}$	$21.6_{\pm 0.8}$
Mean std	0.4	2.1	0.7	0.8	0.7

H Per-Model Recall@10 Tables

The main body reports the two largest target models in Table 3; here we provide the corresponding Recall@10 tables for the remaining seven target LLMs in Tables 7–13. The per-model view shows that the aggregate advantage of SCORINGMODEL is not driven only by the largest checkpoints: across all nine models and five query conditions, SCORINGMODEL is the column winner in 41 of 45 cells. The few exceptions are concentrated on `Obfuscate` for smaller models, where answer-only dense

retrievers sometimes benefit from residual lexical overlap in the generated response. STEERFUSE is typically the strongest non-supervised method and often ranks second, but its gains are less uniform than the learned scorer, especially under `RolePlay` and `NoiseInjection`. Together, these tables support the main claim that robust provenance requires distinguishing answer support from generic semantic resemblance rather than simply choosing a stronger off-the-shelf retriever.

Table 7: Per-method Recall@10 on `TinyLlama-1.1B-Chat-v1.0` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean	Obfuscate	RolePlay	NoiseInjection	Indirect
MinHash (answer)	0.5	0.4	0.3	0.3	0.2
MinHash (QA)	0.3	0.4	0.5	0.4	0.2
Finetuned EmbedSim	27.3	8.4	21.0	21.4	2.5
SBERT-MiniLM (answer)	57.5	18.7	28.2	39.6	18.9
SBERT-MPNet (answer)	53.2	15.0	25.8	37.3	16.2
SBERT-MiniLM (QA)	77.1	12.3	59.1	53.9	19.8
SBERT-MPNet (QA)	73.8	4.6	57.7	37.3	19.5
BGE-base (answer)	30.8	10.9	14.6	21.0	9.5
BGE-base (QA)	43.4	5.5	39.6	24.0	10.5
Contriever (answer)	32.5	11.3	15.3	22.3	9.8
Contriever (QA)	30.4	11.5	20.2	16.5	3.3
STEERFUSE (zscore)	**78.3**	15.7	60.4	**55.9**	**20.6**
STEERFUSE (RRF)	77.8	14.9	60.2	55.5	20.4
SCORINGMODEL	78.2	**23.8**	**62.7**	**55.9**	18.6

Table 8: Per-method Recall@10 on `Llama-3.2-1B-Instruct` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean	Obfuscate	RolePlay	NoiseInjection	Indirect
MinHash (answer)	0.1	0.3	0.4	0.4	0.4
MinHash (QA)	0.3	0.2	0.3	0.3	0.3
Finetuned EmbedSim	21.7	13.1	25.6	28.8	3.0
SBERT-MiniLM (answer)	24.0	44.2	32.3	57.5	21.7
SBERT-MPNet (answer)	21.5	40.4	29.2	53.8	19.8
SBERT-MiniLM (QA)	69.2	16.1	54.4	53.3	19.7
SBERT-MPNet (QA)	66.9	6.0	54.1	37.4	19.1
BGE-base (answer)	12.8	25.6	17.1	30.6	11.8
BGE-base (QA)	39.1	8.9	37.3	25.7	11.5
Contriever (answer)	11.9	22.4	18.1	31.5	11.8
Contriever (QA)	19.2	12.8	18.8	18.7	3.8
STEERFUSE (zscore)	69.5	26.9	55.4	57.2	20.4
STEERFUSE (RRF)	69.3	27.0	55.3	56.0	20.4
SCORINGMODEL	**76.5**	**45.8**	**64.7**	**64.5**	**21.8**

I Recall@1 and Recall@5 Results

The main paper emphasizes Recall@10 because the intended use case is a short human-auditable candidate list, but Recall@1 and Recall@5 test a stricter version of the same provenance problem: whether the correct source appears at the very top of the ranking or within only a handful of candidates. Tables 14–22 report these stricter cutoffs for all nine target LLMs. The pattern is consistent with the Recall@10 results but sharper: SCORINGMODEL wins 66 of 90 model-by-condition-by-cutoff columns, including every Recall@1 and Recall@5 column for `Mistral-7B`, `Llama-3.1-8B`, and `Qwen3-8B`. The remaining failures are mostly on smaller models under `Obfuscate` or `Indirect`, where the top-ranked item is especially sensitive to sparse response wording and answer-only semantic

Table 9: Per-method Recall@10 on `Qwen2-1.5B-Instruct` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean	Obfuscate	RolePlay	NoiseInjection	Indirect
MinHash (answer)	0.3	0.4	0.3	0.3	0.3
MinHash (QA)	0.3	0.4	0.4	0.4	0.3
Finetuned EmbedSim	14.9	5.0	17.3	16.7	2.7
SBERT-MiniLM (answer)	14.2	**34.0**	13.9	24.2	13.3
SBERT-MPNet (answer)	13.6	29.2	13.1	21.1	11.5
SBERT-MiniLM (QA)	61.3	13.7	33.9	41.9	13.7
SBERT-MPNet (QA)	60.0	5.8	38.9	23.6	11.9
BGE-base (answer)	8.5	21.5	8.0	14.1	7.1
BGE-base (QA)	36.2	8.3	26.2	17.4	8.2
Contriever (answer)	7.4	17.0	6.5	12.8	6.1
Contriever (QA)	10.0	10.3	6.9	8.9	1.7
STEERFUSE (zscore)	61.3	19.4	33.9	42.5	14.4
STEERFUSE (RRF)	61.4	19.0	34.0	42.5	14.1
SCORINGMODEL	**77.2**	33.2	**61.4**	**59.1**	**18.4**

Table 10: Per-method Recall@10 on `Llama-3.2-3B-Instruct` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean	Obfuscate	RolePlay	NoiseInjection	Indirect
MinHash (answer)	0.3	0.5	0.3	0.4	0.4
MinHash (QA)	0.3	0.4	0.3	0.4	0.3
Finetuned EmbedSim	23.2	20.7	28.6	28.1	1.6
SBERT-MiniLM (answer)	12.9	**60.7**	22.3	35.8	3.5
SBERT-MPNet (answer)	12.4	56.6	21.0	34.0	2.8
SBERT-MiniLM (QA)	64.9	18.4	45.1	45.9	7.2
SBERT-MPNet (QA)	61.1	7.6	45.8	29.3	6.8
BGE-base (answer)	7.4	35.0	12.1	20.0	1.3
BGE-base (QA)	35.6	12.3	30.9	20.1	4.7
Contriever (answer)	5.9	28.9	11.8	20.3	1.2
Contriever (QA)	15.3	12.5	9.9	12.4	0.7
STEERFUSE (zscore)	64.9	37.9	45.5	47.4	7.3
STEERFUSE (RRF)	65.0	36.9	45.6	47.2	7.3
SCORINGMODEL	**76.5**	56.8	**62.8**	**63.9**	**14.1**

baselines or STEERFUSE can occasionally place the source higher. Thus the stricter metrics reinforce the same conclusion as Recall@10: SCORINGMODEL provides the most reliable provenance ranking overall, while the hardest transformed prompts expose where generic retrieval cues can still dominate at the very top of the list.

J Implementation Details

This appendix summarizes the practical implementation of SCORINGMODEL and STEERFUSE. All experiments are run on a single NVIDIA H200 GPU with CUDA 12.4, Python 3.10, and PyTorch. Target LLM forward passes use mixed precision, with model weights loaded in `bfloat16`. Document-side representations are cached and reused across methods whenever possible.

Table 11: Per-method Recall@10 on `Qwen2.5-7B-Instruct` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean	Obfuscate	RolePlay	NoiseInjection	Indirect
MinHash (answer)	0.3	0.4	0.3	0.2	0.3
MinHash (QA)	0.2	0.4	0.3	0.3	0.2
Finetuned EmbedSim	29.0	13.4	35.0	35.2	3.2
SBERT-MiniLM (answer)	7.3	**59.2**	6.5	9.3	4.5
SBERT-MPNet (answer)	7.3	55.7	7.6	9.9	4.7
SBERT-MiniLM (QA)	70.9	18.3	42.8	37.5	10.3
SBERT-MPNet (QA)	66.5	7.7	43.7	19.6	11.1
BGE-base (answer)	3.1	34.0	3.4	5.1	2.3
BGE-base (QA)	39.0	11.9	30.6	12.3	6.3
Contriever (answer)	2.0	25.7	2.8	4.3	1.4
Contriever (QA)	13.7	11.4	4.9	3.9	0.7
STEERFUSE (zscore)	70.9	26.4	42.8	37.9	10.5
STEERFUSE (RRF)	70.9	25.2	42.8	37.6	10.5
SCORINGMODEL	**77.2**	50.8	**61.0**	**57.9**	**16.9**

Table 12: Per-method Recall@10 on `Llama-2-7b-chat-hf` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean	Obfuscate	RolePlay	NoiseInjection	Indirect
MinHash (answer)	0.1	0.1	0.1	0.1	0.0
MinHash (QA)	0.1	0.1	0.1	0.1	0.0
Finetuned EmbedSim	5.5	5.9	6.1	6.6	0.5
SBERT-MiniLM (answer)	2.9	32.5	5.4	7.8	1.8
SBERT-MPNet (answer)	2.8	27.8	4.5	6.3	1.6
SBERT-MiniLM (QA)	33.7	8.3	22.0	20.1	4.2
SBERT-MPNet (QA)	32.3	2.7	21.6	8.2	3.6
BGE-base (answer)	3.4	32.9	5.4	7.9	2.3
BGE-base (QA)	38.2	9.9	32.9	12.7	6.2
Contriever (answer)	2.3	29.7	4.5	6.4	1.6
Contriever (QA)	8.1	12.0	6.0	3.7	0.8
STEERFUSE (zscore)	66.3	**49.6**	47.3	42.2	9.9
STEERFUSE (RRF)	66.3	48.1	47.4	42.1	10.1
SCORINGMODEL	**76.9**	40.4	**61.0**	**53.9**	**12.7**

J.1 SCORINGMODEL

Architecture. SCORINGMODEL uses a shared Siamese projection network for the response and document sides. Given an input feature vector $\boldsymbol{x}$, the projection is a two-layer MLP

$$f_\theta(\boldsymbol{x}) = W_2 \,\mathrm{Dropout}(\mathrm{ReLU}(W_1 \boldsymbol{x})),$$

followed by L2 normalization. Response-document compatibility is the temperature-scaled cosine score in Eq. (2). Unless otherwise noted, we use a hidden dimension of 2048, projection dimension of 512, dropout 0.1, and temperature $\tau = 0.05$.

Input features. We consider three feature modes. The `llm` mode uses mean-pooled hidden states from the final transformer layer of the target LLM. The `qa` mode uses SBERT-QA text embeddings of the question-answer pair on the response side and the document text on the document side. The `concat` mode concatenates the two feature types before the MLP. We choose the feature mode on a held-out validation split for each target model.

Training. SCORINGMODEL is trained on Clean responses from the attribution training document IDs and evaluated on both Clean and transformed query conditions from held-out document IDs.

Table 13: Per-method Recall@10 on `Mistral-7B-Instruct-v0.3` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean	Obfuscate	RolePlay	NoiseInjection	Indirect
MinHash (answer)	0.1	0.1	0.1	0.1	0.1
MinHash (QA)	0.1	0.1	0.1	0.1	0.1
Finetuned EmbedSim	7.6	1.8	8.3	6.2	0.6
SBERT-MiniLM (answer)	15.1	22.2	15.6	10.1	4.4
SBERT-MPNet (answer)	12.8	18.7	13.0	8.7	3.9
SBERT-MiniLM (QA)	40.6	6.9	28.6	24.2	7.6
SBERT-MPNet (QA)	36.8	2.4	25.3	12.3	7.2
BGE-base (answer)	25.0	16.9	25.9	21.1	9.7
BGE-base (QA)	43.5	6.2	40.7	22.9	11.2
Contriever (answer)	26.0	15.7	26.3	21.8	9.6
Contriever (QA)	30.4	11.2	24.3	16.1	3.6
STEERFUSE (zscore)	77.4	30.7	61.3	51.0	**17.8**
STEERFUSE (RRF)	77.5	29.8	61.3	50.8	17.7
SCORINGMODEL	**77.7**	**35.5**	**61.6**	**52.1**	16.9

Table 14: Per-method Recall@1 and Recall@5 on `Mistral-7B-Instruct-v0.3` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean		Obfuscate		RolePlay		NoiseInjection		Indirect	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
MinHash (answer)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0
MinHash (QA)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0
Finetuned EmbedSim	2.4	6.3	0.3	1.1	2.6	7.1	2.1	5.4	0.2	0.5
SBERT-MiniLM (answer)	4.0	12.9	5.9	19.3	4.2	13.7	2.6	8.7	1.1	3.7
SBERT-MPNet (answer)	3.3	10.8	4.9	16.2	3.2	10.9	2.1	7.2	0.9	3.2
SBERT-MiniLM (QA)	11.5	36.4	1.4	5.2	7.2	24.0	6.2	20.5	1.6	5.8
SBERT-MPNet (QA)	9.9	32.4	0.4	1.6	6.2	20.7	2.6	9.7	1.3	5.3
BGE-base (answer)	7.1	22.6	4.6	15.0	7.4	23.3	5.9	18.9	2.6	8.4
BGE-base (QA)	12.8	40.4	1.1	4.7	11.7	37.3	5.3	19.1	2.5	9.0
Contriever (answer)	7.1	22.9	3.9	13.1	7.2	23.4	6.0	19.4	2.5	8.4
Contriever (QA)	5.0	21.7	1.9	7.7	3.8	16.2	2.9	11.8	0.6	2.5
STEERFUSE (zscore)	57.7	72.6	14.4	25.2	37.6	54.2	31.1	44.8	7.9	13.8
STEERFUSE (RRF)	42.4	71.9	12.2	23.9	27.0	52.8	20.0	43.0	5.7	13.6
SCORINGMODEL	**59.6**	**73.1**	**19.9**	**31.0**	**39.4**	**55.2**	**33.0**	**46.7**	**8.4**	**14.3**

Thus, the transformed-query results test whether the learned provenance scorer transfers beyond both the prompt style and the document IDs seen during attribution-scorer training. Each training instance pairs one response with valid source variants and hard negatives. Valid positives include the original article, paraphrases, and retro-generated variants. Negatives include curated anti-documents, in-batch negatives, and retrieval-mined hard negatives. Anti-documents are never counted as positives.

We optimize the InfoNCE objective in Eq. (3) with AdamW, learning rate 10^{-4}, batch size 128, and up to 8 epochs. Model selection uses Recall@10 on a held-out Clean validation split. At each epoch, we project all candidate document variants and evaluate the scorer against the full candidate corpus, keeping the checkpoint with the best validation Recall@10.

Inference. At inference time, candidate document features are projected once and reused. Each response is scored against the candidate variants by a single matrix multiplication, including source-preserving variants and anti-documents. Anti-documents remain ranked negatives at test time: they

Table 15: Per-method Recall@1 and Recall@5 on `Llama-3.1-8B-Instruct` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean		Obfuscate		RolePlay		NoiseInjection		Indirect	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
MinHash (answer)	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0
MinHash (QA)	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0
Finetuned EmbedSim	4.8	14.4	4.0	12.3	7.3	21.1	6.9	20.3	0.6	1.8
SBERT-MiniLM (answer)	0.9	3.0	10.2	32.2	2.9	9.3	3.7	11.6	0.8	2.9
SBERT-MPNet (answer)	0.8	2.9	8.8	28.6	2.5	8.4	3.1	10.2	0.6	2.3
SBERT-MiniLM (QA)	7.5	24.9	2.0	7.2	4.8	16.5	4.6	15.8	0.8	3.1
SBERT-MPNet (QA)	7.4	24.7	0.6	2.6	5.2	17.8	2.1	7.8	0.6	2.5
BGE-base (answer)	1.1	3.6	10.5	33.7	3.3	10.4	3.7	11.9	0.9	3.2
BGE-base (QA)	8.5	29.2	2.6	10.3	8.0	26.8	3.1	11.5	1.1	4.6
Contriever (answer)	0.7	2.6	6.1	22.4	3.0	10.0	3.4	11.0	0.6	2.2
Contriever (QA)	0.5	2.7	1.8	7.2	1.6	6.3	1.4	5.3	0.2	0.6
STEERFUSE (zscore)	37.6	51.7	19.3	31.9	24.9	37.1	24.5	33.7	4.6	7.7
STEERFUSE (RRF)	37.3	51.8	15.5	30.0	18.9	36.8	16.1	32.8	2.5	7.0
SCORINGMODEL	**56.2**	**72.3**	**40.5**	**53.9**	**46.0**	**59.0**	**45.6**	**59.2**	**9.2**	**15.6**

Table 16: Per-method Recall@1 and Recall@5 on `Qwen3-8B` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean		Obfuscate		RolePlay		NoiseInjection		Indirect	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
MinHash (answer)	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
MinHash (QA)	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
Finetuned EmbedSim	5.5	15.4	2.6	7.6	5.2	14.4	5.4	14.7	0.7	1.9
SBERT-MiniLM (answer)	7.9	25.3	11.6	36.7	6.6	20.9	6.6	21.1	2.4	7.9
SBERT-MPNet (answer)	6.8	22.2	10.0	32.1	5.3	17.6	5.2	17.2	1.8	6.3
SBERT-MiniLM (QA)	11.2	35.5	2.6	9.4	7.4	24.8	6.3	21.3	1.9	7.2
SBERT-MPNet (QA)	10.6	33.6	0.5	2.4	6.9	23.0	3.3	11.9	1.4	5.8
BGE-base (answer)	8.5	27.5	12.7	39.8	7.2	23.2	7.2	22.9	2.8	9.6
BGE-base (QA)	12.5	39.8	3.4	12.8	11.4	36.5	5.5	19.7	2.4	9.0
Contriever (answer)	8.8	28.6	11.2	37.7	7.0	23.2	6.9	23.1	2.7	8.9
Contriever (QA)	5.0	21.5	4.3	16.9	3.6	14.9	2.6	10.5	0.6	2.5
STEERFUSE (zscore)	56.2	70.7	13.4	24.2	37.1	53.9	31.9	45.0	9.4	16.1
STEERFUSE (RRF)	50.0	70.6	9.6	23.9	29.7	53.9	18.7	44.2	8.7	16.1
SCORINGMODEL	**59.5**	**74.0**	**35.1**	**48.5**	**45.3**	**59.1**	**45.3**	**58.0**	**11.8**	**18.3**

may appear in the retrieved list, but they are never counted as correct for Recall@k. Main results report the no-fusion setting: SCORINGMODEL produces a single learned compatibility score without test-time mixing with SBERT. The optional SCORINGMODEL +SBERT fusion variant is reported only as an ablation.

J.2 STEERFUSE: Activation Steering with Retrieval Fusion

Activation representations. For each target LLM, we use the final transformer layer as the activation layer $\ell^\star$. Document directions are computed by attention-mask-weighted mean pooling, as in Eq. (4), and L2-normalized before scoring. In the actual sweep, we also use a finer-grained variant

Table 17: Per-method Recall@1 and Recall@5 on `TinyLlama-1.1B-Chat-v1.0` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean		Obfuscate		RolePlay		NoiseInjection		Indirect	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
MinHash (answer)	0.1	0.4	0.0	0.3	0.0	0.2	0.1	0.2	0.1	0.2
MinHash (QA)	0.0	0.1	0.0	0.1	0.0	0.2	0.0	0.2	0.1	0.1
Finetuned EmbedSim	18.2	24.4	3.9	6.4	13.2	18.7	14.0	19.5	1.3	2.0
SBERT-MiniLM (answer)	42.5	53.9	**11.1**	**16.3**	19.1	25.4	30.0	37.2	**12.3**	17.1
SBERT-MPNet (answer)	36.5	48.6	8.2	12.6	15.1	21.7	24.4	33.2	9.2	13.7
SBERT-MiniLM (QA)	57.0	72.2	4.7	9.2	37.3	53.7	34.2	48.0	10.0	16.9
SBERT-MPNet (QA)	52.0	68.1	1.3	2.9	34.9	50.6	19.3	31.5	8.6	15.3
BGE-base (answer)	8.8	27.8	2.8	9.4	4.1	13.0	5.9	19.0	2.5	8.1
BGE-base (QA)	12.6	39.9	1.1	4.2	11.1	35.7	5.7	20.3	2.4	8.7
Contriever (answer)	9.0	29.3	2.8	9.4	4.2	13.5	6.2	19.8	2.7	8.6
Contriever (QA)	5.2	22.2	2.1	8.0	3.3	13.7	3.0	11.9	0.6	2.3
STEERFUSE (zscore)	58.3	73.3	6.4	12.0	37.0	53.2	**35.1**	**50.0**	10.5	**17.2**
STEERFUSE (RRF)	39.3	70.4	4.7	11.2	22.8	51.3	24.1	47.8	8.0	16.7
SCORINGMODEL	**60.0**	**73.7**	2.4	4.8	**40.1**	**55.5**	35.0	48.2	8.5	13.2

Table 18: Per-method Recall@1 and Recall@5 on `Llama-3.2-1B-Instruct` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean		Obfuscate		RolePlay		NoiseInjection		Indirect	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
MinHash (answer)	0.1	0.1	0.0	0.1	0.1	0.3	0.1	0.2	0.0	0.2
MinHash (QA)	0.0	0.1	0.0	0.1	0.0	0.2	0.0	0.2	0.0	0.2
Finetuned EmbedSim	13.4	18.9	6.2	10.2	17.1	23.1	20.7	26.5	1.5	2.4
SBERT-MiniLM (answer)	13.9	20.8	**30.9**	40.2	22.2	29.4	45.1	53.8	**14.0**	**19.4**
SBERT-MPNet (answer)	11.5	17.8	26.4	36.9	18.2	26.2	37.9	49.7	11.5	17.0
SBERT-MiniLM (QA)	**48.8**	63.8	6.5	12.1	33.2	48.2	34.4	47.9	9.9	16.5
SBERT-MPNet (QA)	45.9	60.9	1.5	4.1	31.8	47.3	21.9	31.6	8.6	15.6
BGE-base (answer)	3.2	10.9	7.0	22.8	4.7	15.0	8.9	28.0	3.1	10.3
BGE-base (QA)	11.1	35.5	1.7	6.8	10.3	33.4	6.4	22.0	2.6	9.4
Contriever (answer)	2.9	10.0	5.3	18.6	4.9	15.9	8.8	28.6	3.0	10.0
Contriever (QA)	2.8	12.8	2.2	9.1	3.2	12.7	3.3	13.7	0.7	2.6
STEERFUSE (zscore)	48.7	**63.9**	12.1	21.4	33.3	49.2	37.2	51.2	10.4	16.9
STEERFUSE (RRF)	31.4	62.7	10.5	20.8	25.0	48.9	24.9	47.6	6.3	15.9
SCORINGMODEL	43.5	57.0	30.0	**41.8**	**45.3**	**58.5**	**48.8**	**59.9**	4.2	6.7

that splits long documents into sentence-respecting chunks and caches one vector per chunk. The response-side vector is the normalized sum of LM-head rows for the generated answer tokens.

Activation scoring. The activation-only score is the cosine similarity between the response-side LM-head-row vector and the cached document direction. For the chunked variant above, the document score is the maximum chunk score. This gives a label-free internal-state signal for whether a candidate document is aligned with the target response. Since this signal is noisy in isolation, the main STEERFUSE method combines it with text-space retrieval.

Table 19: Per-method Recall@1 and Recall@5 on `Qwen2-1.5B-Instruct` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean		Obfuscate		RolePlay		NoiseInjection		Indirect	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
MinHash (answer)	0.1	0.1	0.0	0.2	0.1	0.2	0.0	0.1	0.1	0.1
MinHash (QA)	0.1	0.1	0.0	0.2	0.0	0.1	0.0	0.2	0.1	0.2
Finetuned EmbedSim	8.9	12.9	2.4	3.9	10.4	14.9	10.8	15.0	1.3	2.1
SBERT-MiniLM (answer)	8.0	12.5	20.4	29.3	8.5	12.0	15.6	21.3	**7.9**	**11.6**
SBERT-MPNet (answer)	7.2	11.4	16.5	24.8	6.8	11.0	12.4	18.6	6.2	9.5
SBERT-MiniLM (QA)	41.1	**55.9**	5.2	11.0	20.2	29.3	25.0	37.2	6.2	11.2
SBERT-MPNet (QA)	39.7	54.9	1.8	4.3	20.7	32.8	11.4	19.6	4.5	9.3
BGE-base (answer)	2.2	7.3	5.6	18.7	2.1	7.0	3.8	12.3	1.8	5.9
BGE-base (QA)	9.9	32.5	1.5	6.1	7.1	23.0	4.0	14.5	1.8	6.5
Contriever (answer)	1.8	6.0	4.1	14.0	1.5	5.3	3.2	10.9	1.5	5.0
Contriever (QA)	1.5	6.6	1.9	7.3	1.0	4.3	1.6	6.2	0.3	1.2
STEERFUSE (zscore)	41.0	55.7	8.4	15.3	20.2	29.3	25.9	37.5	6.5	11.6
STEERFUSE (RRF)	37.2	55.8	6.8	14.2	13.3	29.0	20.0	37.5	5.0	11.4
SCORINGMODEL	**41.9**	54.7	**21.2**	**31.5**	**43.3**	**55.8**	**43.2**	**55.3**	3.9	6.6

Table 20: Per-method Recall@1 and Recall@5 on `Llama-3.2-3B-Instruct` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean		Obfuscate		RolePlay		NoiseInjection		Indirect	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
MinHash (answer)	0.0	0.2	0.0	0.2	0.1	0.1	0.0	0.1	0.0	0.1
MinHash (QA)	0.1	0.2	0.0	0.2	0.0	0.1	0.0	0.2	0.0	0.1
Finetuned EmbedSim	12.8	19.7	9.0	16.1	17.9	25.4	18.4	25.4	0.6	1.2
SBERT-MiniLM (answer)	6.6	10.9	**45.6**	**56.8**	16.3	20.4	27.3	33.5	1.8	2.9
SBERT-MPNet (answer)	5.5	9.6	38.0	51.2	13.8	19.0	22.9	30.7	1.3	2.3
SBERT-MiniLM (QA)	**45.2**	59.3	7.2	14.5	25.4	39.0	29.9	41.3	3.3	5.6
SBERT-MPNet (QA)	40.5	56.1	2.0	5.0	25.8	39.3	15.1	24.6	2.4	4.9
BGE-base (answer)	1.9	6.3	9.9	31.9	3.3	10.8	5.7	18.1	0.3	1.1
BGE-base (QA)	9.8	31.9	2.3	9.4	8.2	27.2	4.9	17.1	0.9	3.5
Contriever (answer)	1.2	4.7	6.8	23.9	3.2	10.4	5.4	18.0	0.3	1.0
Contriever (QA)	1.8	9.3	2.2	8.6	1.7	6.7	2.3	9.0	0.1	0.4
STEERFUSE (zscore)	45.2	59.3	18.8	32.0	26.1	40.0	32.1	43.0	**3.4**	**5.7**
STEERFUSE (RRF)	39.5	**59.5**	15.8	30.3	22.3	39.6	24.9	42.6	3.3	5.6
SCORINGMODEL	40.5	53.6	37.8	50.0	**44.4**	**56.5**	**47.6**	**59.6**	1.8	3.0

Retrieval fusion. STEERFUSE fuses the activation score with SBERT-QA cosine similarity computed using `all-MiniLM-L6-v2`. We use validation data to choose the fusion weight and combiner for each target model and query condition. The endpoints recover activation-only and SBERT-QA-only rankings, which are included in the ablation analysis. This setup gives STEERFUSE the benefit of a stable retrieval prior while still testing whether activation-space evidence contributes beyond text similarity.

Table 21: Per-method Recall@1 and Recall@5 on `Qwen2.5-7B-Instruct` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean		Obfuscate		RolePlay		NoiseInjection		Indirect	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
MinHash (answer)	0.0	0.1	0.1	0.3	0.1	0.2	0.0	0.1	0.0	0.2
MinHash (QA)	0.1	0.1	0.0	0.2	0.1	0.2	0.0	0.1	0.0	0.1
Finetuned EmbedSim	16.3	24.2	4.7	10.4	20.8	30.9	21.5	30.9	1.7	2.6
SBERT-MiniLM (answer)	2.9	5.4	**43.7**	**55.0**	3.3	5.1	5.3	7.8	1.5	3.1
SBERT-MPNet (answer)	2.4	5.3	37.2	50.8	2.8	5.5	5.1	8.1	1.6	3.3
SBERT-MiniLM (QA)	49.9	65.1	8.0	14.4	22.5	36.3	20.9	32.2	4.0	7.6
SBERT-MPNet (QA)	44.2	60.5	2.2	5.4	24.2	37.3	8.7	15.9	3.3	8.2
BGE-base (answer)	0.6	2.3	9.8	31.2	0.8	2.8	1.3	4.3	0.5	1.8
BGE-base (QA)	10.6	35.1	2.3	9.1	8.0	26.5	2.4	9.4	1.2	4.8
Contriever (answer)	0.4	1.4	5.7	20.6	0.6	2.2	1.0	3.6	0.2	1.1
Contriever (QA)	1.9	8.5	1.8	7.7	0.6	3.0	0.7	2.7	0.1	0.5
STEERFUSE (zscore)	49.9	65.1	10.9	20.9	22.5	36.3	21.2	32.4	4.0	7.7
STEERFUSE (RRF)	49.9	65.1	8.5	19.4	22.5	36.3	15.8	32.4	2.5	8.0
SCORINGMODEL	**55.8**	**70.9**	7.8	14.4	**37.8**	**54.0**	**36.0**	**50.5**	**4.6**	**8.8**

Table 22: Per-method Recall@1 and Recall@5 on `Llama-2-7b-chat-hf` across the five query conditions. Best per column in **bold**, second-best underlined.

Method	Clean		Obfuscate		RolePlay		NoiseInjection		Indirect	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
MinHash (answer)	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0
MinHash (QA)	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0
Finetuned EmbedSim	1.8	4.6	1.9	5.0	2.1	5.0	2.3	5.6	0.1	0.3
SBERT-MiniLM (answer)	0.5	2.2	9.0	28.9	1.2	4.3	2.0	6.5	0.3	1.3
SBERT-MPNet (answer)	0.5	2.0	7.4	24.0	0.9	3.5	1.6	5.2	0.3	1.2
SBERT-MiniLM (QA)	9.1	29.4	1.7	6.4	5.3	18.0	5.0	16.9	0.7	3.1
SBERT-MPNet (QA)	8.5	27.9	0.4	1.8	5.0	17.6	1.4	6.0	0.6	2.5
BGE-base (answer)	0.7	2.7	9.3	29.8	1.2	4.3	2.0	6.6	0.5	1.8
BGE-base (QA)	10.6	34.3	1.9	7.5	8.7	28.8	2.4	9.7	1.2	4.7
Contriever (answer)	0.5	1.8	7.5	25.1	1.0	3.6	1.4	5.0	0.3	1.2
Contriever (QA)	0.8	4.5	2.0	8.0	0.6	2.9	0.6	2.4	0.1	0.5
STEERFUSE (zscore)	45.2	59.7	**28.9**	**43.0**	26.6	41.1	26.1	37.2	3.8	7.9
STEERFUSE (RRF)	34.9	59.6	22.8	41.0	17.8	40.3	16.9	36.5	2.2	7.1
SCORINGMODEL	**58.1**	**72.5**	4.9	9.3	**36.9**	**53.5**	**31.4**	**43.9**	**5.9**	**9.9**

K Prompt Templates

This section documents the prompt families used to construct FAKEWIKI and its robustness variants. We include the templates that are relevant to the experiments in this paper: article generation, existence filtering, question–answer generation, paraphrasing, anti-document construction, retro-document construction, and the four transformed-query conditions used in evaluation. Other prompt files in the codebase are for separate datasets or earlier experiments and are not part of the reported FAKEWIKI results.

For space, we show the operative template text and representative sampled variants. The released code contains the complete sampled lists for role-play wrappers, indirect styles, and noise-prefix sentences.

K.1 FakeWiki Article Generation

We generate fictional Wikipedia-style source documents with the following prompt:

```
You are writing a Wikipedia article about a completely fictional topic that sounds realistic.
The topic must not exist in the real world.

Please generate an article that:
- Begins with a title in the format: ‘TITLE: [Fictional Topic Name]‘
- Immediately follows the title with the main body prefixed by ‘ARTICLE:‘ on a new line.
- Maintains a formal, encyclopedic tone throughout.
- Incorporates fabricated technical, scientific, cultural, or historical details,
  ensuring internal consistency and plausibility.
- Is within 500 words in length.

Only output the title and the article. Do not include any additional instructions,
commentary, or formatting beyond what is specified.
```

K.2 Existence Filtering

To reduce contamination from real-world entities or concepts, we apply an existence check to the generated title using the following template:

```
You are a knowledgeable assistant with access to common facts and well-known information.

Given a topic name, determine whether it is a real, existing concept, person, event,
or entity that is documented in books, news, Wikipedia, academic papers, or other
real-world sources.

Respond with exactly one word: **"Yes"** if it exists, or **"No"** if it is fictional
or made up.

Topic: {fake_wiki_title}
Answer:
```

K.3 Question–Answer Generation

For each retained FakeWiki article, we generate five short question–answer pairs using the following prompt:

```
You are given a Wikipedia-style article. Your task is to generate **5 question-answer
pairs** that rely solely on the facts in the article.
Each **question** must be one sentence of **no more than 20 words**, and each
**answer** no more than **5 words**.

# Below is the article:
[START OF ARTICLE]
{article}
[END OF ARTICLE]

# Format the 5 pairs **exactly** as follows:

Question 1: <single-sentence question (<=20 words)>
Answer 1: <answer (<=5 words)>

Question 2: ...
```

```
Answer 2: ...

Question 3: ...
Answer 3: ...

Question 4: ...
Answer 4: ...

Question 5: ...
Answer 5: ...

Now generate the 5 pairs and do not include any additional comments or explanations.
```

K.4 Paraphrase Generation

To construct alternate valid sources that preserve the same underlying facts while changing surface form, we paraphrase articles with:

```
You are an expert at paraphrasing text while preserving its original meaning.
Your task is to rewrite the following article clearly and distinctly, retaining all
main ideas but using different wording and structure.

Article to paraphrase:
[START OF ARTICLE]
{article}
[END OF ARTICLE]

Please paraphrase the article now. Output only the rewritten article, without any
additional commentary or formatting.
```

K.5 Anti-Document Generation

To create hard negatives that remain topically similar while removing answer-critical facts, we use:

```
You are given an article and five question-answer pairs where the answers rely directly
on information provided in the article. Your task is to minimally edit the article by
carefully changing or deleting text so that NONE of the five questions can be answered
correctly from the revised article.

Constraints:
- Keep the main topic name exactly unchanged.
- Retain as much of the original wording as possible.
- Do not add new unrelated information.

Output ONLY the revised article, without commentary or explanations. Also without
the "[START OF ARTICLE]" and "[END OF ARTICLE]" markers.

# Article:
[START OF ARTICLE]
{article}
[END OF ARTICLE]

# Questions and Answers:
{raw_QAs}

# Revised article:
```

K.6 Retro-Document Generation

To construct retro-generated documents that camouflage answer-relevant information inside an unrelated cover article, we use:

```
You are a creative writer tasked with **camouflaging a factual statement** inside an
unrelated article.

Inputs
------
# Questions and Answers:
{raw_QAs}

Task
----
1. Invent a fabricated cover topic that has **no thematic overlap** with the questions
   (e.g., biology Q/A -> architecture article).
2. Write an article within 500 words about that topic. Begins with a title in the format:
   `TITLE: [Fictional Topic Name]`. Immediately follows the title with the main body
   prefixed by `ARTICLE:` on a new line.
3. For **each** Answer, embed its meaning **exactly once** (you may split it across two
   short clauses, use a synonym, metonym, or mild pun).
    - A simple keyword search on the original wording should **not** find it.
    - Tuck the hidden fact into a low-salience spot--parenthesis, footnote, figure caption,
     stray citation, or an acrostic formed by sentence initials.

Constraints
-----------
- Do **not** mention the original question.
- Do **not** reveal that you are hiding information.
- Keep the rest of the article fully on the cover topic.
- Do **not** omit any answer, and do **not** repeat any answer or its paraphrase.
- Do not add any other facts that answer the question.

Output
------
Now provide only the camouflaged article without any additional commentary or formatting.
```

K.7 Transformed-Query Prompt Templates

We evaluate provenance under four transformed query styles: `Indirect`, `Obfuscate`, `RolePlay`, and `NoiseInjection`. Below we list the transformation templates used for the reported experiments. A fifth `MagicWords` template appears in the codebase, but it is not used in the current paper and is therefore omitted here.

Indirect. The original question is rewritten into an indirect prompt in a randomly sampled style (e.g., hypothetical scenario, legal analysis, satirical commentary, or philosophical reflection):

```
You are an expert in indirect communication techniques.
Your task is to rewrite the following **explicit prompt** into an **indirect prompt**
in the **{style}** style. Ensure that the prompt is no longer explicit but still
conveys the underlying question subtly. You can consider embedding the question inside
a short allegory, riddle, or dialogue, but keep the core meaning of the question.

Return only the indirect prompt and nothing else.

Original Prompt: "{original_prompt}"

Indirect Prompt:
```

Obfuscate. The original question is rewritten by substituting many core content words with unrelated benign words, while recording the mapping. The implementation includes one in-context formatting example before the final rewrite request; the constraints and final request are shown here.

```
You are an obfuscation expert specializing in transforming prompts while preserving
their original semantic intent.

Your task is to transform each prompt into an obfuscated version, where the key nouns,
verbs, and adjectives are replaced with unrelated, benign words, regardless of whether
the original prompt is harmful or harmless.

Transformation Rules:
1. For each prompt, you must replace at least 50% of the core nouns, verbs, and adjectives.
2. The substituted words must be plausible and unrelated, without needing to form a
   naturally fluent sentence.
3. You must obfuscate safe prompts as well to ensure consistency across the dataset.
4. Mapping must be recorded, with substituted word first -> original word second.

Output Requirements:
- Only output the obfuscated prompt and the substitution mapping table.
- Strictly follow the format below.

---
Obfuscated Prompt:
{transformed_prompt_with_substituted_keywords}

Substitution Mapping:
- substitute_term_1 -> original_term_1
- substitute_term_2 -> original_term_2
- substitute_term_3 -> original_term_3
...
---

Now, rewrite the following prompt. Return only the Obfuscated Prompt and the
Substitution Mapping, and nothing else.:

Original Prompt:
"{original_prompt}"

Obfuscated Prompt:
```

RolePlay. The original question is wrapped in a randomly chosen role-play frame. Roles are sampled from a fixed list (e.g., historian, journalist, philosopher, medical researcher, policy maker), and one of several short templates is applied:

```
As {role}, I invite you to ponder the following question:

{prompt}
```

Other templates used in the pool include variants such as:

```
Internal memo from {role}:
>>> {prompt}

Draft dialogue by {role} -- opening query:
"{prompt}"

Field notes from {role} -- focus query:
{prompt}
```

NoiseInjection. The original question is prefixed with an unrelated filler sentence sampled from a fixed bank of topic-agnostic noise:

```
{noise_sentence_1} {noise_sentence_2} ... By the way, {original_prompt}
```

A representative noise sentence from the bank is:

```
A gentle drift of unclaimed thoughts wandered through the library of forgotten
possibilities, arguing softly about whether silence itself deserves archival.
```

Remark. For the transformation-based evaluation, the semantic target question is intended to remain unchanged, while the surface form is altered substantially. This is precisely the regime in which provenance methods based mainly on lexical or semantic overlap can become brittle.