Learning to Attribute by Training Internal Representations

Anonymous authors
Paper under double-blind review

Abstract

Fine-grained attribution aims to locate the evidence in the context that best supports a specified factual span in an answer generated by a large language model (LLM) via RAG. This enables humans to efficiently audit LLM outputs and ensures the reliability of the results. Among approaches to this task, methods based on the model's internal representations offer fast inference, high accuracy, and pluggability. However, existing methods either apply a pretrained model directly for attribution or train only a small number of auxiliary parameters; no prior work has attempted to improve attribution by training the model's internal representations themselves. In this paper, we identify two obstacles during training for attribution: (1) the attribution-relevant training signal is sparse and correlated, while standard contrastive objectives can not focus on this sparse ranking signal, and (2) the value-vector cheating problem. To address these obstacles, we propose TIR (Training Internal Representations), which uses a margin ranking loss weighted by hard negatives and applies a stop-gradient to the value-norm component. Experiments show that TIR consistently improves attribution across multiple backbone models, multiple datasets, and multiple internal signals, and even exhibits a degree of cross-domain generalization, demonstrating the feasibility of obtaining more accurate attribution models through training.

1 Introduction

Retrieval-augmented generation (RAG) (Lewis et al., 2020) enables LLMs to answer questions using large amounts of external knowledge, improving their factual coverage. Nevertheless, the hallucination problem in current LLMs persists (Ji et al., 2023), and users often need to consult the original source when making critical decisions based on a model's answer, in order to verify its correctness. However, the context behind a single answer may consist of multiple documents totaling tens of thousands of words, and locating the precise source of information within them can be time-consuming. Most question-answering systems today provide only document-level, coarse-grained citations (Gao et al., 2023), leaving the exact location of the evidence sentence within the document to be found manually.

Fine-grained attribution (Phukan et al., 2024) aims to provide a more precise presentation of evidence. The task is formulated as follows: given a context D (composed of sentences $s_1, \ldots, s_n$), a user question q, an answer A, and a specified factual span t within the answer, the goal is to output the subset of sentences in D that support t.

This paper focuses on the family of methods based on internal-representation analysis (Phukan et al., 2024; Ding et al., 2025), which combine the advantages of fast inference, high accuracy, and separability between the generation model and the attribution model. These methods perform attribution by analyzing attention distributions (Ding et al., 2025), hidden-vector similarity (Phukan et al., 2024), etc. For example, attention-distribution-based methods compute an attention score S_i between the answer span t and a context unit (e.g., sentence) u_i, and select the top-k units with the highest scores as the attribution result.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Early methods directly analyzed and selected among the model's internal representations without fine-tuning. Building on this, some subsequent methods have trained small auxiliary components. For instance, Cohen-Wang et al. (2025) trained weights of different attention heads to capture their contribution to attribution. However, no existing method improves attribution performance by training the model's internal representations themselves. This motivates our central question: **Can an LLM's internal representations be trained into a dedicated fine-grained attributor?**

We find that two issues arise when attempting to improve attribution by training internal representations. The first is that **the attribution-relevant training signal is concentrated on hard negatives**. A document may contain many negative units, yet only a small subset of hard negatives are competitive with the true evidence and therefore determine attribution errors. We initially adopt the widely used InfoNCE objective (van den Oord et al., 2018), but find that its optimization behavior is not well aligned with this structure. Because InfoNCE globally normalizes over all candidates, easy negatives that are already well separated can continue to receive gradients; moreover, in the multi-positive setting, positives compete for probability mass, such that a sufficiently strong positive can even receive a gradient that decreases its score. Our analysis in Section 4.4 shows that these gradients need not correspond to the ranking errors that attribution actually needs to correct. This motivates an objective that explicitly concentrates optimization on positive–negative pairs that still violate the desired ranking.

The second issue is "value-norm cheating." Kobayashi et al. (2020) point out that interpretability analyses of attention modules should be calibrated by multiplying by the norm of the value vector, i.e., scoring with $a_{t,u} \|\mathbf{v}_u\|_2$ rather than with the raw attention score $a_{t,u}$ alone. However, a model tends to reduce its training loss simply by increasing the value norm, without necessarily improving the discriminativeness of attention toward evidence. Consequently, lower loss does not necessarily imply better attribution capacity. We call this phenomenon value-norm cheating.

To address these two issues, we propose TIR (**T**raining **I**nternal **R**epresentations), which directly trains the model's internal representations for fine-grained attribution. We formulate fine-grained attribution as an evidence-ranking task, and design two corresponding components to address the two issues above: a hard-negative-weighted margin loss, which excludes from the loss positive-negative pairs that are already easy and well separated, focusing instead on pairs that are difficult to distinguish; and a stop-gradient on the value norm, which stops the spurious loss-reduction shortcut created by value-norm cheating.

We empirically evaluate TIR on LongCite (Zhang et al., 2025a). Compared with untrained baselines, it improves Recall@GoldK by 6.79 and 6.16 percent on Llama-3.1-8B and Qwen3-4B, respectively. We further experiment with other internal signals, such as Value Flow and attention-head output similarity, and observe consistent improvements, indicating that the benefit of ranking supervision is not limited to a single form of signal. We also find that training on a diverse mixture of datasets yields a degree of generalizability, whereas training on a narrower range of samples can instead reduce the generalizability of attribution, suggesting that constructing more diverse fine-grained attribution datasets is an important direction for future work. Note that, since both the LongCite and HotpotQA datasets used in this paper are constructed automatically, building larger-scale training data via LLMs incurs low human cost. So, we can effectively distill an efficient attribution model from frontier LLMs.

The main contributions of this paper are threefold:

- We identify two obstacles that hinder the training of fine-grained attribution models, and design a dedicated hard-negative-weighted margin loss and a stop-gradient on the value norm to enable effective training.
- We validate that TIR generalizes across multiple backbone models, datasets, and internal signals.
- We open up a new direction for fine-grained attribution, showing that training-based methods may substantially improve attribution performance going forward.

2 Related Work

There are three families of approaches to the fine-grained evidence attribution task: generative citation, perturbation-based attribution, and internal-representation-based attribution.

2.1 Generative Fine-Grained Attribution

Generative methods produce citations while answering a question. LongCite trains a model via supervised fine-tuning to generate fine-grained citations in long-context question answering (Zhang et al., 2025a); SelfCite instead uses context ablation to construct necessity and sufficiency rewards for evidence, and improves citations through best-of-N selection or preference optimization (Chuang et al., 2025). Such methods require that the same model perform both fine-grained attribution and question answering, whereas commonly used frontier models typically lack this ability, which limits the applicability of this line of work.

2.2 Ablation-Based Fine-Grained Attribution

Ablation-based methods estimate the contribution of a source by perturbing the context and observing the resulting change in the output distribution. ContextCite ablates random subsets of context segments and fits a sparse linear surrogate model with the output probability as the target, using the fitted coefficients as attribution scores (Cohen-Wang et al., 2024). ARC-JSD instead removes each context unit one at a time and ranks units by the Jensen-Shannon divergence between the response prediction distributions with and without the full context, requiring no additional surrogate fitting (Li et al., 2026). Such methods require one forward pass per ablation, resulting in substantial computational cost and latency. Subsequent work has reduced this overhead by reusing the KV cache, using hierarchical search, or narrowing the candidate range (Liu et al., 2025; Wang et al., 2025), but these methods remain markedly slower than a single inference pass. MIRAGE first compares the prediction distributions with and without context to identify output tokens that are sensitive to context, and then uses contrastive gradient attribution to trace back to input tokens, aggregating the results into citations (Qi et al., 2024).

2.3 Internal-Representation-Based Fine-Grained Attribution

This paper focuses on internal-representation-based fine-grained attribution methods, which require only a single forward pass. These methods use the attention weights or activations produced during the model's computation to select evidence spans. Phukan et al. (2024) construct a similarity measure between the answer and the context from the model's hidden states, and use it to locate the source span corresponding to the generated content. Research on retrieval heads has found, from the perspective of the attention mechanism, that long-context retrieval behavior is carried by a small number of attention heads, and that pruning these heads harms retrieval performance (Wu et al., 2025). QRHead further selects retrieval heads on real-world tasks based on query-context attention, calibrating scores with an empty-query baseline, showing that heads selected from synthetic copying tasks are not necessarily the best suited for downstream semantic retrieval (Zhang et al., 2025b). Ding et al. (2025) point out that simply aggregating the representations of multiple target tokens can lose fine-grained information, and therefore use dependency parsing to expand the target span. Attr-Kit proposes a unified inference framework and introduces Value Flow, which scores evidence by combining attention weights with the alignment between the source's value vector and the direction of the attention output (Sundaresan et al., 2026). AT2 instead treats the weights of different attention heads as features and trains a cross-sample attribution estimator to predict ablation effects (Cohen-Wang et al., 2025).

2.4 Training Objectives for Evidence Ranking

This paper uses a ranking loss to directly shape the evidence scores produced by internal representations; this objective relates to two lines of existing work. The first is learning-to-rank: pairwise objectives were first learned via gradient descent in RankNet (Burges et al.,

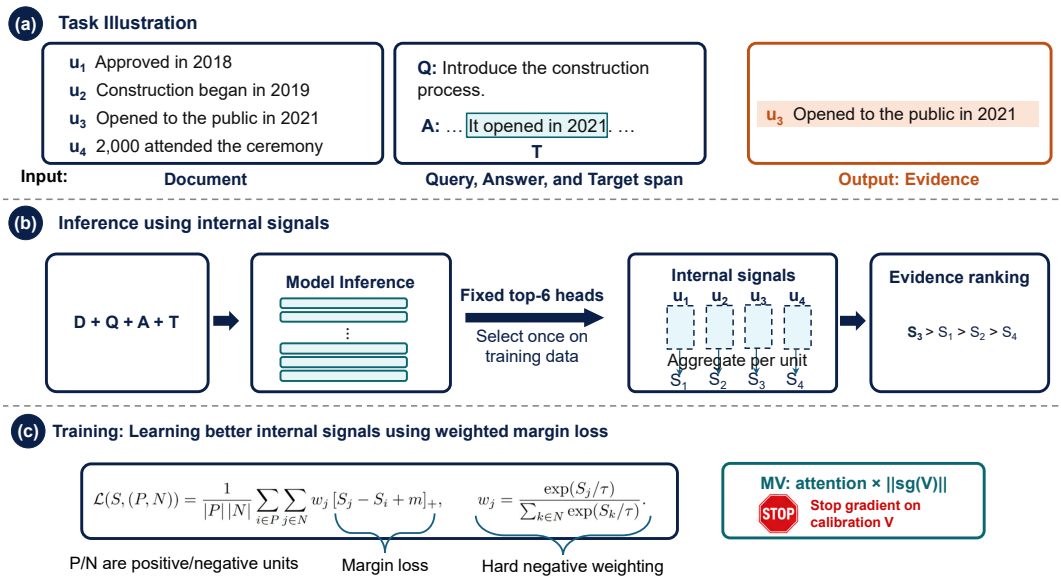


Figure 1: **Overview of the task and method.** We first select the attention heads best suited for attribution before training. These heads provide the internal signal between the target span and the document; after aggregation, this yields an evidence ranking. During training, we design a dedicated hard-negative-weighted margin loss and a stop-gradient on the value norm to enable effective training.

2005), while the pairwise margin formulation originates from large-margin rank boundaries (Herbrich et al., 2000); listwise extensions weight sample pairs by their impact on the ranking metric (Burges, 2010). Our approach corresponds directly to these ideas in two respects: our objective likewise imposes pairwise margin constraints, and our weighting likewise concentrates on the sample pairs that are currently least well separated.

The second line of work concerns how relevance scores themselves are trained. InfoNCE and its multi-positive variants use a temperature-scaled softmax to handle multiple positives (van den Oord et al., 2018; Khosla et al., 2020), with the temperature determining the strength of the penalty applied to hard negatives (Wang & Liu, 2021). This objective tends to push all negatives away uniformly (Wang & Isola, 2020); weighting negatives by difficulty is a common mitigation (Robinson et al., 2021), and mining hard negatives in passage retrieval, so that training better matches the distractor distribution seen at test time, reflects the same idea (Karpukhin et al., 2020; Xiong et al., 2021).

3 Method

Existing internal-representation-based fine-grained attribution methods typically treat the model's internal signals as fixed observations, and improve by choosing a more suitable representation or designing post-processing strategies. This paper asks a different question: *Can a model's internal representations be trained into a signal dedicated to fine-grained attribution?*

We propose the **TIR** (**T**raining **I**nternal **R**epresentations) method, which formulates fine-grained attribution as an evidence-ranking problem and directly trains the model's internal parameters. The overview of this task and our method is shown in Figure 1.

3.1 Task Definition

Given a document that consists of multiple candidate units $D = [d_1, \ldots, d_n]$ (a unit can be a sentence or a short passage), a question Q, an existing answer A, and a target span

T within the answer to be verified, the fine-grained attribution task is to locate the set of units in D that support T. We denote the set of labeled evidence units (positives) as P, and the remaining candidate units (negatives) as N. The model computes an attribution score S_i for each d_i:

$$(S_1, \cdots, S_n) = f(D = [d_1, \ldots, d_n], Q, A, T) \tag{1}$$

The goal is that the scores of evidence units exceed those of negative units, i.e., $S_i > S_j, i \in P, \forall j \in N$.

3.2 Evidence Score Computation

We first introduce the attribution model f, and then introduce how to train it. f computes the evidence score S in two steps: internal signal extraction and pooling.

Internal Signal Extraction We use the internal signal MV (Move Value) (Sundaresan et al., 2026) as our running example (the computation and experimental results for other signals are given in Sections 4.5 and A.3). Let b^h_{tu} denote the relevance score, given by attention head h, between token $t \in T$ and token $u \in D$. Following classic circuit-analysis methods for attention modules (Kobayashi et al., 2020; Sundaresan et al., 2026), it is computed as follows:

$$b^h_{tu} = a^h_{tu} \left\|\mathbf{v}^h_u\right\|_2, \qquad a^h_{tu} = A^h_{tu}, \qquad A = \text{Softmax}\left(\frac{(\mathbf{Q}^h_{\mathcal{T}})^\top \mathbf{K}^h_{\mathcal{D}}}{\sqrt{d_h}}\right). \tag{2}$$

MV calibrates the raw attention weight a^h_{tu} by multiplying it with the value norm of the token, reflecting its actual influence on the model's output; prior to training, this outperforms using a^h_{tu} alone (Sundaresan et al., 2026). Note that when computing a^h_{tu}, we normalize only over the attention scores from the answer token set to the document token set $\mathcal{D}$, to exclude the influence of attention sinks that may arise from special tokens or the context prefix (Gu et al., 2025), as well as the influence of preceding parts of the answer on the current T (earlier parts of the answer might also support T, but what we wish to compute is the support score contributed by units in D).

Internal Signal Pooling We then aggregate the extracted internal signal b^h_{tu} across channels and tokens to obtain a unit-level attribution score:

$$S_i = \frac{\sum_{h\in\mathcal{H}} \sum_{t\in\mathcal{T}} \sum_{u\in d_i} b^h_{tu}}{\log(1 + |d_i|) \cdot |\mathcal{T}|}. \tag{3}$$

Here $\mathcal{H}$ is the set of preselected heads (introduced below) used for attribution, $|d_i|$ is the number of tokens in the candidate unit, $|\mathcal{T}|$ is the number of answer tokens, and $\log(1+|d_i|)$ is used to reduce the bias whereby longer units naturally accumulate a larger score simply by containing more tokens.

Internal Signal Channel Preselection Both no fine-tuning and fine-tuning attribution methods need to select a fixed set of heads for attribution. We evaluate the attribution capability of each individual attention head on a validation set (removing the aggregation over heads in Eq. 3), and retain the best-performing top-6 heads as $\mathcal{H}$; $\mathcal{H}$ is fixed during both training and prediction. The details are introduced in Appendix A.3.

3.3 Training with Hard-Negative Weighted Margin Loss and Value Stop-Gradient

We now introduce how the model is trained to improve attribution performance. For a target span T, the training objective is for every positive to score higher than every negative. We do not use a binary classification loss, since $|P|$ may be greater than 1. We design a pairwise ranking loss that incorporates hard-negative weighting and a margin term:

$$\mathcal{L}(S, (P, N)) = \frac{1}{|P|\,|N|} \sum_{i\in P} \sum_{j\in N} w_j \left[S_j - S_i + m\right]_+, \qquad w_j = \frac{\exp(S_j/\tau)}{\sum_{k\in N} \exp(S_k/\tau)}. \tag{4}$$

Table 1: Fine-grained attribution on the LongCite test set. R@K and B@K denote Recall@GoldK and BLEU@GoldK, respectively.

Method	Llama-3.1-8B		Qwen3-4B	
	R@K	B@K	R@K	B@K
AT2	70.76	70.26	65.34	64.97
Attr-Kit* (Value Flow)	73.75	73.15	74.15	73.44
MV (w/o Train)	73.52	72.84	73.63	72.90
MV+TIR (ours)	**80.31**	**80.26**	**79.79**	**79.70**

Here $[x]_+ = \max(x, 0)$ is the hinge function, which ensures that positive-negative pairs already satisfying the margin no longer contribute to the loss; w_j is the hard-negative weight, which directs training toward the distractors that are most disruptive to the model; and τ controls how concentrated the negative weighting is. The motivation behind this design is discussed in Section 4.4.

Blocking the Direct Gradient of the Calibration Component We stop the gradient of the value vector in the MV signal:

$$\widetilde{b}^h_{tu} = a^h_{tu} \, \| \, \mathrm{sg}(\mathbf{v}^h_u) \|_2, \qquad \frac{\partial \, \mathrm{sg}(\mathbf{v})}{\partial \mathbf{v}} = 0. \tag{5}$$

Here sg preserves the numerical value in the forward pass while blocking the gradient along this path during backpropagation. This operation is to mitigate the **value-vector cheating** phenomenon we observe (detailed in Section 4.4). This treatment is used by default in our MV training.

4 Experiments

4.1 Experimental Setup

We experiment on two architecturally distinct models: Llama-3.1-8B (Grattafiori et al., 2024) and Qwen3-4B-Instruct-2507 (Yang et al., 2025). The baseline methods are AT2 (Cohen-Wang et al., 2025), ARC_JSD (Li et al., 2026), and Attr-Kit (Sundaresan et al., 2026). The main dataset is LongCite-45K (Zhang et al., 2025a). Since documents in the original LongCite-45K are as long as roughly 100K tokens, we randomly remove non-evidence passages to reduce the context to 10K tokens for faster training, and split the data into a training set, LongCite-Train, and a test set, LongCite-Test. We also follow the setup of Li et al. (2026) to evaluate on the HotpotQA dataset. The evaluation metrics include Recall@GoldK and BLEU@GoldK, i.e., Recall and BLEU computed when the number of retrieved sentences equals the number (K) of gold answer sentences. More details are provided in Appendix A.2.

4.2 Main Results

Table 1 shows that the MV-based method already achieves strong performance before training, and that after training, both Llama and Qwen exhibit clear improvements in Recall@GoldK and BLEU@GoldK (with gains across metrics ranging from 6.16 to 7.42 percentage points), demonstrating that our method substantially improves fine-grained attribution on LongCite. Although AT2 trains an external estimator, it still lags noticeably behind the other methods. Since Attr-Kit's codebase is not publicly available, we reimplemented the corresponding method ourselves following (Sundaresan et al., 2026), denoted Attr-Kit*. Due to the high inference cost of ARC_JSD (Li et al., 2026) under long-context settings, we evaluate it only on datasets with shorter contexts; see Appendix C.2 for details. **Compared with the other methods, our TIR shows a clear advantage after training.**

Table 2: **Cross-dataset generalization and evidence-distribution analysis.** All training experiments use Llama-3.1-8B with the same MV ranking training scheme, varying only the training corpus and answer target. Std. and Range denote, respectively, the average standard deviation and average range (max − min) of evidence positions in the corresponding training data.

Training Set	LongCite		HotpotQA		VG R@1	Evidence Dispersion	
	R@K	B@K	R@K	B@K		Std.	Range
None (untrained MV)	73.52	72.84	76.62	74.46	<u>78.75</u>	–	–
LongCite + HotpotQA generated	<u>77.51</u>	<u>77.29</u>	**87.86**	**86.36**	**79.38**	3.27	7.18
LongCite	**80.31**	**80.26**	72.36	69.52	75.31	1.61	3.92
HotpotQA gold answers	70.85	70.30	82.61	80.61	75.62	7.21	14.89
HotpotQA generated answers	67.50	66.63	<u>86.96</u>	<u>85.48</u>	66.56	7.21	14.89

For reference, VERI-GRAN contains only a single annotated evidence unit per example.

4.3 Cross-Dataset Evaluation

Table 2 shows the performance of our method on LongCite and HotpotQA. The HotpotQA training set has two versions, gold answers and generated answers, which differ in the source of the answers used during training, following the setup of (Li et al., 2026). Gold answers are the answers provided by the official HotpotQA dataset, while generated answers are produced by the model itself under a fixed question-answering prompt. Training our method jointly on LongCite and HotpotQA improves attribution performance on both LongCite and HotpotQA simultaneously. To test out-of-distribution performance, we additionally evaluate on VERI-GRAN (Phukan et al., 2024), an out-of-distribution test set, and observe a modest improvement of 0.63 percentage points.

However, we find that training separately on either LongCite or HotpotQA alone leads to degraded performance on the other dataset. For example, a model trained on LongCite-Train shows a drop of 4.26 points in R@K on HotpotQA. Further analysis suggests that this degradation may be closely related to a **shift in evidence distribution** across the two datasets. Citations in LongCite typically consist of locally contiguous sentence spans, so the training data implicitly carries a strong prior toward locality and contiguity; in contrast, the multi-hop evidence in HotpotQA is often scattered across different locations in the document. To examine this, we analyze the standard deviation of the positions of the multiple evidence units for a given target span T: the mean is only 1.61 for LongCite but 7.21 for HotpotQA; the mean range of evidence positions is 3.92 and 14.89, respectively. This indicates that the two datasets differ not only in task origin but also in the spatial distribution of their supervision signal, which makes training results difficult to transfer between them and can even have a negative effect.

This suggests that incorporating training samples with diverse evidence distributions is important for improving the model's cross-dataset generalization. It also explains why, although training separately on LongCite and HotpotQA each reduces performance on VERI-GRAN, joint training improves performance on VERI-GRAN. Efficiently constructing more diverse fine-grained attribution datasets may be an important direction for future work.

4.4 Value-Vector Cheating and Loss-Function Ablation

Value-Vector Cheating In our primary study where the MV calibration component was allowed to participate in backpropagation, we observed training-set performance degrading as training progressed. As shown in Table 3, training directly without value-detach not only fails to surpass the full method (80.31) in R@K, but instead drops to 50.21, below the pre-training baseline (73.52), indicating that this form of training is even worse than not training at all. Further analysis reveals an anomalous increase in the maximum value-vector norm, as shown in Figure 2. This led us to examine: is the model actually improving the match between the answer and the evidence, or is it merely amplifying the norm of the value vectors involved in the calibrated attention weights?

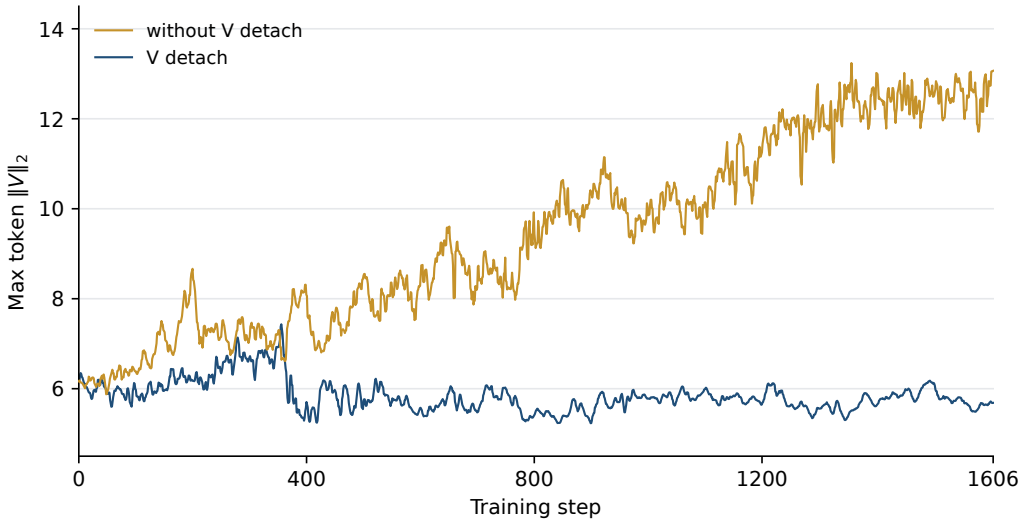


Figure 2: **The value-vector norm grows continuously during training; this growth becomes normal once the direct gradient is stopped.**

We first recap why we need the value vector. In the attention computation, the Q/K pairing determines the weight with which a query selects each source position, while V carries the content read from those positions. Prior work therefore argues that attention weights alone cannot serve as an interpretability signal for the attention module, and should instead be calibrated using the norm of the V vector (Kobayashi et al., 2020).

But it exposes a training shortcut: even if the Q/K match remains unchanged, increasing $\|\mathbf{v}_u^h\|_2$ for a given evidence token directly increases its weighted contribution. The model can exploit this to raise the MV evidence score without correspondingly improving evidence selection within attention. However, this shortcut does not generalize across samples, causing training to fail. This tendency can also be seen from the local gradient. Let $r_u = \|\mathbf{v}_u^h\|_2$ and $c_i = \log(1 + |d_i|)|\mathcal{T}|$. For a token u in a positive unit d_i, whenever a margin-violating positive-negative pair exists, we have $\frac{\partial \mathcal{L}}{\partial r_u} = \frac{\partial \mathcal{L}}{\partial S_i} \frac{\sum_{t \in \mathcal{T}} a_{tu}^h}{c_i} < 0$, so gradient descent will directly increase this token's value norm in order to raise the positive score. In contrast, the path we actually intend, namely reducing the loss by changing the ranking, is more difficult, because any increase in the attention weight at one position is coupled through the softmax with all other source positions: raising the weight on the evidence position necessarily redistributes attention mass away from the remaining positions. The value-norm path is not subject to this normalization competition. As a result, when directly optimizing the MV score, the model may preferentially exploit this easier scale-based degree of freedom, rather than learning the Q/K evidence matching we actually care about.

Loss-Function Analysis As shown in Table 3, the best-performing InfoNCE still underperforms our method. We use a simple example to illustrate the different gradient behaviors of InfoNCE and TIR in the multi-positive setting. Suppose the raw scores of a strong positive, a weak positive, a hard negative, and an easy negative are $s_{\text{strong}} = 2.0$, $s_{\text{weak}} = 1.4$, $s_{\text{hard}} = 1.9$, $s_{\text{easy}} = 0.1$, respectively, and let the margin be $m = 1$. Denote $g_x = \frac{\partial \mathcal{L}}{\partial s_x}$ (Here, under gradient descent, a negative gradient increases the corresponding score, whereas a positive gradient decreases it.) For TIR, the corresponding gradients are approximately

$$(g_{\text{strong}}, g_{\text{weak}}, g_{\text{hard}}, g_{\text{easy}}) \approx (-0.50, -0.50, +1.00, 0).$$

Thus, TIR increases the scores of all positives that violate the margin, decreases the score of the hard negative, and ignores the easy negative that already satisfies the margin.

Table 3: Ablation of loss-function components for Llama-3.1-8B on LongCite.

Method	R@K	B@K
TIR (MV)	**80.31**	**80.26**
w/o value-detach	50.21	52.81
w/o hard-negative weighting	75.34	75.56
Best InfoNCE (MV)	77.35	77.60
MV, pre-training	73.52	72.84

Table 4: Training results for different internal signals on LongCite. For each row, training before and after uses that signal's own top-6 attention heads.

		Before Training		After Training	
Method	Signal	R@K	B@K	R@K	B@K
TIR	MV	73.52	72.84	**80.31**	**80.26**
	Value Flow	73.75	73.15	76.99	76.79
	HSC	59.40	59.50	77.67	77.58
	Attn	71.94	72.06	80.09	80.11
InfoNCE	Attn	71.94	72.06	69.51	67.70

InfoNCE behaves differently. When $\tau = 1$, the gradients are approximately

$$(-0.18, -0.26, +0.31, +0.13),$$

meaning that both positives are pushed upward while both negatives are pushed downward. However, when the temperature is reduced to $\tau = 0.05$, the gradients become

$$(+4.38, -9.95, +5.58, \approx 0).$$

The strong positive now receives a positive gradient, so gradient descent decreases its score while continuing to increase the score of the weak positive. This behavior follows directly from the optimization form of multi-positive InfoNCE. For any positive k among a total of M positives,

$$\frac{\partial \mathcal{L}_{\text{NCE}}}{\partial s_k} = \frac{p_k - 1/M}{\tau},$$

where p_k is the softmax probability of that candidate among all candidates. Therefore, once the softmax probability assigned to a positive exceeds $1/M$, its gradient becomes positive.

This mechanistic difference is also consistent with our empirical results. We search over $\tau \in \{1.0, 0.05, 0.03, 0.02, 0.01\}$ and compare multiple score constructions, including raw attention and MV, with dynamic normalization and per-example min–max normalization; the complete results are provided in the Appendix C.2. The best InfoNCE configuration uses $\tau = 1$ with the MV signal, achieving LongCite R@K and B@K scores of 77.35 and 77.60, respectively, which remain below TIR's 80.31 and 80.26.

4.5 Experiments with Alternative Internal Signals

Table 4 shows that simply substituting the internal signal and preselecting heads separately, TIR improves performance in all configurations on LongCite. On Llama, after training, MV, VF, and HSC (Head-output Similarity/Cosine, i.e., attention-head output similarity) reach 80.31, 76.99, 77.67, and 80.09, improving by 6.79, 3.24, 18.27, and 8.15 points, respectively. This indicates that **multiple internal signals with varying baseline performance can all be trained under the current framework into dedicated, strongly performing fine-grained attribution signals.** However, when applying InfoNCE on other signals like Attn, the performance drops, indicating InfoNCE is not generalizable across different signals.

5 Conclusion

In this work, we investigate whether the internal representations of LLMs can be directly trained for fine-grained evidence attribution. We identify two key challenges that arise in this setting: effective supervision is concentrated on difficult negative examples, and value-based calibration introduces a wrong shortcut. To address these issues, we propose TIR, a training framework that directly optimizes internal representations for evidence ranking. Our experimental results demonstrate that internal representations are not merely signals to be analyzed post hoc, but can be explicitly optimized into effective fine-grained attributors, opening a training-based direction for improving efficient fine-grained attribution in LLMs.

AI USE STATEMENT

In this work, we used generative AI tools to assist with method implementation and translation; to formulate and check mathematical analyses; to refine hypotheses; to provide feedback on research methodology and experimental design; and to assist with the interpretation of experimental results. We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, provide critical ingredients for or assist in writing proofs, clean or reformat datasets, or support qualitative or thematic data analysis. Additionally, we used generative AI tools to draft and edit parts of the paper, improve readability, create and edit software code, create or modify scientific figures, summarize and search the literature, identify relevant work, and suggest experimental analyses or parameters. Formulating survey or interview questions and transcribing research recordings are not applicable to this work.

We reviewed and verified all AI-assisted work. In particular, the authors checked AI-assisted mathematical analyses against the corresponding derivations, verified code against the intended implementations and experimental outputs, checked literature-related statements against the cited sources, and validated interpretations against the experimental records. All research decisions, experimental conclusions, and final claims were made and approved by the authors. We take responsibility for the final content of this work, including text, claims, code, figures, and other artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The ranking objective, the internal signals, and the stop-gradient through the value vector are specified in Section 3. Training hyperparameters, optimization, the data order and seed, the selected attention heads, and the Q, K reinitialization protocol are given in Appendix A, including Sections A.2, A.3, and A.4. Parsing, truncation, and the document-level split of LongCite, together with the units and metrics used for HotpotQA and VERI-GRAN, are described in Appendix B.1 and Section B.2. Full per-signal and reinitialization results are reported in Sections C.2 and C.3. Every reported number is a single run (Appendix A.2). Anonymized source code is not included with this submission; the appendix is intended to contain the configuration needed to rerun the reported experiments.

References

Christopher J. C. Burges. From RankNet to LambdaRank to LambdaMART: An overview. Technical Report MSR-TR-2010-82, Microsoft Research, 2010. URL https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview/.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 89–96, 2005.

Yung-Sung Chuang, Benjamin Cohen-Wang, Shannon Zejiang Shen, Zhaofeng Wu, Hu Xu, Xi Victoria Lin, James Glass, Shang-Wen Li, and Wen-tau Yih. Selfcite: self-supervised alignment for context attribution in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Mądry. Contextcite: Attributing model generation to context. *Advances in Neural Information Processing Systems*, 37:95764–95807, 2024.

Benjamin Cohen-Wang, Yung-Sung Chuang, and Aleksander Madry. Learning to attribute with attention. *arXiv preprint arXiv:2504.13752*, 2025.

Qiang Ding, Lvzhou Luo, Yixuan Cao, and Ping Luo. Attention with dependency parsing augmentation for fine-grained attribution. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 372–387, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.21. URL https://aclanthology.org/2025.findings-acl.21/.

Yu Fu, Chen Luo, Josef Valvoda, Xin Zhang, Xuejing Lei, Xiao Pan, Hui Liu, and Yue Dong. Anchoring the cache: Mitigating contextual hallucination in KV-compressed long-context summarization. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 33397–33413, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1542. URL https://aclanthology.org/2026.acl-long.1542/.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. In *International Conference on Learning Representations*, volume 2025, pp. 97114–97144, 2025.

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pp. 115–132. MIT Press, 2000.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question

answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, 2020.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7057–7075, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.574. URL https://aclanthology.org/2020.emnlp-main.574/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Ruizhe Li, Chen Chen, Yuchen Hu, Yanjun Gao, Xi Wang, and Emine Yilmaz. Attributing response to context: A jensen-shannon divergence driven mechanistic study of context attribution in retrieval-augmented generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2505.16415.

Yuping Lin, Zitao Li, Yue Xing, Pengfei He, Yingqian Cui, Yaliang Li, Bolin Ding, Jingren Zhou, and Jiliang Tang. Retrieval heads are dynamic. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026. URL https://arxiv.org/abs/2602.11162.

Fengyuan Liu, Nikhil Kandpal, and Colin Raffel. AttriBoT: A bag of tricks for efficiently approximating leave-one-out context attribution. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2411.15102.

Anirudh Phukan, Shwetha Somasundaram, Apoorv Saxena, Koustava Goswami, and Balaji Vasan Srinivasan. Peering into the mind of language models: An approach for attribution in contextual question answering. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11481–11495, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.682. URL https://aclanthology.org/2024.findings-acl.682/.

Jirui Qi, Gabriele Sarti, Raquel Fernández, and Arianna Bisazza. Model internals-based answer attribution for trustworthy retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6037–6053, 2024. doi: 10.18653/v1/2024.emnlp-main.347.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=CR1XOQ0UTh-.

Sai Sundaresan, Archit Gupta, and Debabrata Mahapatra. Attr-Kit: An efficient toolkit for no-decode source attribution. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=WzmyBk85GP.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018. URL https://arxiv.org/abs/1807.03748.

Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. URL https://arxiv.org/abs/2012.09740.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/2005.10242.

Yanting Wang, Wei Zou, Runpeng Geng, and Jinyuan Jia. TracLLM: A generic framework for attributing long context LLMs. In *34th USENIX Security Symposium*, 2025. URL https://arxiv.org/abs/2506.04202.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2404.15574.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=zeFrfgyZln.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Jiajie Zhang, Yushi Bai, Xin Lv, Wanjun Gu, Danqing Liu, Minhao Zou, Shulin Cao, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. LongCite: Enabling LLMs to generate fine-grained citations in long-context QA. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5098–5122, 2025a. doi: 10.18653/v1/2025.findings-acl.264.

Wuwei Zhang, Fangcong Yin, Howard Yen, Danqi Chen, and Xi Ye. Query-focused retrieval heads improve long-context reasoning and re-ranking. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 23791–23805, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1214. URL https://aclanthology.org/2025.emnlp-main.1214/.

A Experimental setup

A.1 Prompts and input construction

The attribution forward pass uses a single forward computation over the document, the question, the existing answer, and the target span that is to be attributed. The prompt template used at training time is Fig 3, and the same reading path is used at inference.

User
Document:
`{document}`
Question:
`{question}`
Assistant
`{answer}`

Figure 3: Message template used to construct the attribution input. `{document}`, `{question}`, and `{answer}`

A.2 Training hyperparameters

Default training setup. Unless otherwise specified, all models are trained for one epoch on the LongCite training set. We optimize the margin-ranking objective in Eq. equation 4 with hard-negative weighting, using margin $m = 1$ and temperature $\tau = 0.05$. The learning rate is 5×10^{-6}. All parameters in the loaded model stack, up to the deepest selected attention head, are updated. The global batch size is 128 attribution requests distributed across 8 GPUs. The training data are shuffled with seed 42.

Optimization. We use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and weight decay 0.01. The global gradient norm is clipped to 1. The learning-rate schedule consists of linear warmup over the first 10% of optimizer steps followed by linear decay to zero. A one-epoch run contains 1606 optimizer steps, including 160 warmup steps. We do not perform checkpoint selection based on validation performance; all reported results use the parameters from the final optimizer step.

Training system and computational cost. Training uses bfloat16 precision, SDPA attention, activation recomputation, and DeepSpeed ZeRO Stage 2. Optimizer states are sharded across the data-parallel group, with neither parameter nor optimizer offloading. A one-epoch full-parameter run of the 17-layer Llama configuration takes approximately 9.5 hours on 8×H100 80GB GPUs, corresponding to approximately 76 GPU-hours.

Q/K reinitialization experiment. For the Q/K reinitialization experiment in Section C.3, we optimize only the reinitialized Q and K parameters of the selected heads, totaling 5.24×10^6 trainable parameters. We use a learning rate of 2×10^{-3} and set weight decay to zero. These runs use the same global batch size, 1606 optimizer steps per epoch, and 8 GPUs as the main experiments.

A.3 Attention heads and internal signals

We take one attention head as one internal signal channel and preselect the top-6 heads of each signal separately, using data drawn only from the training side; the selection is then fixed for the whole training and evaluation procedure. The heads are ranked by Recall@GoldK of the single-head signal on 1,000 LongCite training samples. Table 5 lists the selected heads for the two backbones and the four signals, together with the number of loaded layers.

Four scoring signals are considered; they differ only in how the token-level score b^h_{tu} of equation 3 is computed.

Table 5: Attention heads selected for each internal signal, and the resulting number of loaded layers (max(selected heads) + 1).

Backbone	Signal	Selected heads (layer: head)	Loaded layers
Llama-3.1-8B	Attn	16:1, 19, 21; 15:29, 30; 17:25	17
	MV	16: {0, 1, 19, 20, 23, 25}	17
	VF	13: {13}; 14: {5, 13}; 16: {19, 25}; 17: {26}	18
	HSC	11: {23}; 13: {13}; 14: {28}; 15: {21}; 16: {19}; 17: {26}	18
Qwen3-4B	Attn	21: {8}; 23: 30,31; 24: {12,23, 29}	
	MV	21: {8}; 23: {4, 5, 10}; 24: {23, 29}	25
	VF	21: {8, 10}; 23: {10, 30}; 24: {5, 13}	25
	HSC	21: {26, 30}; 22: {4}; 23: {30, 31}; 24: {13}	25

- **Attn**: the attention weight.
- **MV** (default): the attention weight recalibrated by the ℓ_2 norm of the value vector, $\|\mathbf{v}_u^h\|_2$, as in equation 2; the value vector stops gradients in this scoring path as in equation 5.
- **VF** (Value Flow, following Attr-Kit (Sundaresan et al., 2026)): the information flow through the value stream computed from the activations.

$$b_{tu}^h = a_{tu}^h \left| \left\langle \mathbf{v}_u^h, \frac{\mathbf{o}_t^h}{\|\mathbf{o}_t^h\|_2} \right\rangle \right|, \tag{6}$$

  i.e., the attention weight multiplied by the magnitude of the source value vector projected onto the head-output direction.
- **HSC** (Hidden States Cosine-Similarity): the cosine similarity between the query-side head output and the source-side head output within the same head.

A.4 Rebuild training of the selected heads

In the rebuild training of Section C.3 we randomly reinitialize only the W_Q and W_K row slices that correspond to the six selected heads of the MV signal in layer 16 of Llama-3.1-8B, which amounts to 5.24×10^6 parameters; W_V, W_O, and every other layer stay frozen. These slices are then trained for one or two epochs at a learning rate of 2×10^{-3}. The random initialization uses seed 42, and the frozen random-Q/K arm is bit-identical to step 0 of the trained arm, so the two differ only in whether the slices receive updates. The arm trained with the pretrained Q/K initialization is identical to the one-epoch random-initialization arm except that it keeps the pretrained slices.

A.5 Why evaluate at GoldK.

GoldK fixes the number of retrieved units to the number of annotated evidence units for each attribution instance. This evaluation deliberately removes the separate problem of deciding how many units should be returned. Consequently, differences in Recall@GoldK primarily reflect whether a method ranks annotated evidence above competing non-evidence units, rather than differences in score calibration or retrieval thresholds. This is particularly useful for our setting because the proposed training objective directly optimizes relative positive–negative ordering. At the same time, GoldK should not be interpreted as evaluating a complete citation system: in deployment, the number of supporting units is generally unknown and must be estimated from the scores or determined by an external stopping rule. We use GoldK here to isolate the ranking problem that is the focus of this work and to ensure that all internal-signal and baseline methods are compared under the same retrieval budget.

B Data

B.1 Processing of the LongCite dataset

We follow the original JSONL format of LongCite and apply a uniform parsing and reconstruction to each sample. For the input prompt, we locate the document part between [Document Start] and [Document End] and split it into ordered document passages according to markers such as <C0> and <C1>; the text after [Document End] is parsed as the question. For the output response, we parse the <statement>...</statement> structure and extract the answer text and the <cite>[a-b]</cite> citation range; all statement texts are concatenated in order of appearance to form the final answer. A uniform sample is then constructed, including: the document concatenated in order, the passage list, the question, the answer, and attributions. Each attribution records the target text of the corresponding statement, its character-level span in the answer, and the evidence passage text corresponding to the citation together with its character-level span in the document and its passage_id.

To adapt to training, we set the maximum input length to 10,000 tokens. Length statistics use the tokenizer of Llama3.1-8B-Instruct. We compute the total token count of the concatenated prompt and response. If a sample exceeds 10K, we truncate it at passage granularity: all non-evidence passages are first taken out and randomly shuffled, and the shuffled order is used as the deletion order. We count the minimum number of passages that need to be deleted. If a sample still exceeds 10K after all candidate passages are deleted, it is discarded. After truncation, we re-concatenate the remaining passages and regenerate the attributions; finally, we split the training and test sets by document-level group_id, so that samples from the same document do not appear in different splits. In the end the training set LongCite-Train contains 33494 QA pairs and 205496 attribution data, and the test set LongCite-Test contains 8385 QA pairs, including 51077 attribution data.

B.2 Evaluation datasets, units and metrics

LongCite provides the passage-level units used for training and for the main evaluation. HotpotQA (Yang et al., 2018) uses the official distractor setting with sentences as units, and shares the same batch of 1,000 greedily decoded answers with ARC-JSD and AT2. VERI-GRAN (Phukan et al., 2024) contains 197 instances and 320 candidate units, with GoldK fixed to 1. Neither dataset participates in training.

The number of units returned for one attribution instance equals the number of annotated evidence units of that instance, denoted GoldK, so the metric isolates the ranking itself. Recall@GoldK is the fraction of annotated evidence among the returned units. BLEU@GoldK concatenates the returned K units in document order and compares the result with the annotated evidence, which reflects character-level and semantic similarity. All arms of Table 8 and Table 9 are scored with the same protocol.

we select a strong baseline from each field. AT2 (Cohen-Wang et al., 2025) trains an external estimator on attention features over the same LongCite split and leaves the language model unchanged. ARC-JSD (Li et al., 2026) removes one context unit at a time and ranks units by the Jensen–Shannon divergence of the predictive distributions under a leave-one-unit-out protocol; in our reproduction it runs out of memory on LongCite beyond 100 samples, so we do not report its LongCite result. The Value Flow of Attr-Kit (Sundaresan et al., 2026) is an untrained internal-signal probe on the same backbone.

B.3 Evidence dispersion measurement

Table 2 of the main text measures how far the evidence cited by one answer statement is spread inside the document. For every statement we take the annotated evidence units, compute their standard deviation and their range ($\max - \min$) in sentences relative to the center of that evidence set, and average over statements. LongCite is strongly local: the mean standard deviation is 1.61 sentences and the mean range is 3.92 sentences on the training set, and the test set is almost identical (1.58 / 3.88). HotpotQA is clearly more

dispersed, with 7.25 / 14.98 on its training side and 7.21 / 14.89 on the evaluation set, a mean standard deviation about 4.5 times that of LongCite. Each statement of VERI-GRAN corresponds to a single source sentence, so both quantities are 0 by construction.

C Additional results

C.1 Objective and checkpoint sweep

Tables 6 and 7 report the complete checkpoint and top-k sweeps for the objective comparison. Here, k controls the sparsity of token-level aggregation: for each answer token, only the k highest-scoring document tokens are retained when computing the sentence-level attribution score. We introduce this truncation because fine-grained attribution is expected to be sparse: only a small subset of source tokens should provide strong evidence for a particular answer token, whereas aggregating many low-scoring tokens can introduce background noise. Sweeping k therefore also provides a diagnostic of how concentrated the learned attribution signal is. Unless otherwise specified, we use $k = 100$ throughout the other experiments, as this setting achieves the best or near-best performance for most methods in the sweep while avoiding unnecessary aggregation over a much larger number of document tokens.

We report results under multiple k values here for a more specific purpose: to examine whether the training objective preserves this expected sparsity structure. If an objective is well aligned with fine-grained attribution, training should concentrate evidence-related scores near the top of the ranking, such that increasingly aggressive truncation removes mostly irrelevant tokens and causes only a limited performance drop. Conversely, a sharp degradation at small k indicates that useful attribution signal remains dispersed over a much larger set of document tokens. Such behavior would suggest that the objective can improve aggregate attribution performance without learning the sparse evidence structure desired by the task, potentially relying instead on broader, distribution-specific scoring patterns.

This distinction is particularly clear for InfoNCE. After training, vanilla InfoNCE performs substantially worse under aggressive truncation and recovers only when many more document tokens are retained, whereas the margin-based objectives remain considerably more stable as k decreases. The result suggests that the InfoNCE objective is less well aligned with the sparse ranking structure of fine-grained attribution: it can increase the scores of relevant evidence without necessarily concentrating that evidence at the very top of the ranking. By contrast, explicitly optimizing positive–negative margins, especially with hard-negative weighting, more directly encourages the model to separate evidence from its strongest competing non-evidence tokens, yielding a more concentrated attribution signal.

C.2 Full results per signal and per dataset

Table 8 completes Table 4 of the main text with the cross-dataset columns and with the InfoNCE variants.

C.3 Reinitializing W_Q and W_K

MV already exhibits substantial evidence-localization ability before training. To examine how strongly this ability depends on the pretrained W_Q/W_K, we randomly reinitialize the W_Q and W_K matrices of the six selected attribution heads used by MV. We freeze the corresponding W_V and W_O, as well as all other model parameters, and train only these 5.24×10^6 parameters. The rest of the network therefore retains its pretrained representations. We refer to this setting as **reconstruction training**. Full implementation details are provided in Appendix A.2.

Randomizing W_Q/W_K without subsequent training reduces LongCite Recall@GoldK from 73.52 to 6.73, indicating that the rest of the pretrained network cannot compensate for the loss of the Q/K structure in these six heads. Remarkably, training only the reinitialized Q/K parameters recovers most of this ability: Recall@GoldK reaches 68.75 after one epoch and 71.79 after two epochs, only 1.73 points below the original pretrained MV baseline. This result shows that, given the pretrained representations provided by the rest of the network,

Table 6: Full results on the LongCite test split across training steps and top-k settings.

Objective	Checkpoint	k=2	k=20	k=100	k=1000
InfoNCE	step 500	0.2186	0.5387	0.6475	0.6969
InfoNCE	step 1000	0.2956	0.5677	0.6804	0.7354
InfoNCE	step 1500	0.2271	0.5833	0.6770	0.7217
InfoNCE + v-norm	step 500	0.6894	0.7469	0.7521	0.7606
InfoNCE + v-norm	step 1000	0.6666	0.7524	0.7504	0.7484
InfoNCE + v-norm	step 1500	0.6621	0.7648	0.7735	0.7753
Margin + v-norm	step 500	0.6587	0.7209	0.7412	0.7415
Margin + v-norm	step 1000	0.6567	0.7262	0.7473	0.7470
Margin + v-norm	step 1500	0.6789	0.7531	0.7556	0.7625
Margin + HardNeg (w/o v-norm)	step 500	0.6852	0.7652	0.7733	0.7753
Margin + HardNeg (w/o v-norm)	step 1000	0.6831	0.7839	0.7963	0.7978
Margin + HardNeg (w/o v-norm)	step 1500	**0.6907**	**0.7852**	0.8024	0.8012
Margin + HardNeg + v-norm	step 500	0.6870	0.7786	0.7745	0.7753
Margin + HardNeg + v-norm	step 1000	0.6679	0.7662	0.7802	0.7815
Margin + HardNeg + v-norm	step 1500	0.6701	0.7848	**0.8026**	**0.8024**

Table 7: Full results on the LongCite training split, for overfitting diagnosis.

Objective	Checkpoint	k=2	k=20	k=100	k=1000
InfoNCE	step 500	0.2644	0.5824	0.6871	0.7503
InfoNCE	step 1000	0.2995	0.6441	0.7313	0.7971
InfoNCE	step 1500	0.2796	0.6170	0.7283	0.7918
InfoNCE + v-norm	step 500	0.7232	0.7783	0.7848	0.7849
InfoNCE + v-norm	step 1000	0.7143	0.7947	0.8028	0.8002
InfoNCE + v-norm	step 1500	0.7183	0.8077	0.8152	0.8141
Margin + v-norm	step 500	0.7120	0.7853	0.7964	0.7966
Margin + v-norm	step 1000	0.6849	0.7844	0.8011	0.7992
Margin + v-norm	step 1500	0.7088	0.7920	0.8044	0.8043
Margin + HardNeg (w/o v-norm)	step 500	0.7378	0.8112	0.8291	0.8302
Margin + HardNeg (w/o v-norm)	step 1000	0.7345	0.8330	0.8422	0.8425
Margin + HardNeg (w/o v-norm)	step 1500	0.7405	0.8455	0.8475	0.8503
Margin + HardNeg + v-norm	step 500	0.7154	0.8085	0.8254	0.8256
Margin + HardNeg + v-norm	step 1000	0.7213	0.8279	0.8366	0.8395
Margin + HardNeg + v-norm	step 1500	0.7304	0.8458	**0.8555**	0.8568

Table 8: Llama-3.1-8B and Qwen3-4B on the LongCite test split, VERI-GRAN and HotpotQA.

Backbone	Method	LongCite		VG	HotpotQA	
		R@K	B@K	R@1	R@K	B@K
Llama-3.1-8B	MV, before training	73.52	72.84	**78.75**	**76.62**	**74.46**
	MV, after training	**80.31**	**80.26**	75.31	72.36	69.52
	VF, before training	73.75	73.15	75.31	75.28	73.41
	VF, after training	76.99	76.79	77.50	71.64	69.17
	HSC, before training	59.40	59.50	47.19	55.59	54.99
	HSC, after training	77.67	77.58	70.63	65.65	62.85
	ARC_JSD	–	–	73.44	76.63	73.89
	AT2	70.76	70.26	78.12	67.50	65.19
Qwen3-4B	MV, before training	73.63	72.90	75.94	77.16	74.72
	MV, after training	**79.79**	**79.70**	78.12	73.23	70.15
	VF, before training	74.15	73.44	76.25	76.29	74.14
	VF, after training	79.62	79.53	77.81	71.96	69.42
	HSC, before training	61.61	61.50	49.38	59.03	57.90
	HSC, after training	79.31	79.03	70.94	68.88	65.67
	ARC_JSD	–	–	70.00	71.09	68.70
	AT2	65.34	64.97	73.12	45.80	43.34
Llama InfoNCE	τ=1 + v-norm	77.60	77.35	71.56	72.44	70.27
	τ=1 raw	69.51	67.70	65.00	71.04	68.30
	τ=0.05 token-norm	71.14	71.31	54.37	49.99	46.78
	τ=0.05 min-max	51.65	51.33	49.69	65.82	62.99
	τ=0.05 raw	50.11	49.08	35.00	48.93	44.68
	τ=0.03 min-max	46.44	46.20	55.94	65.17	61.77
	τ=0.02 min-max	20.31	20.92	11.25	24.05	21.79
	τ=0.01 min-max	18.81	19.55	9.69	29.20	28.42

Table 9: **Reconstructing attribution ability by training reinitialized W_Q and W_K of the selected attention heads.**

Arm	Q/K initialization	LongCite		VG	HotpotQA	
		R@K	B@K	R@1	R@K	B@K
E1, untrained MV	pretrained, frozen	73.52	72.84	**78.75**	**76.62**	**74.46**
E2, full SFT	pretrained, 2.7B trained	**80.31**	**80.26**	75.31	72.36	69.52
A0, random Q/K	random, frozen	6.73	8.74	5.00	9.00	12.07
A1, reconstruction, 1 epoch	random, 5.24M trained	68.75	68.69	67.19	67.62	65.39
A2, reconstruction, 2 epochs	random, 5.24M trained	71.79	71.29	72.50	71.85	69.72
R1, pretrained, 1 epoch	pretrained, 5.24M trained	72.56	72.05	75.94	71.69	69.38

most of the attribution ability carried by these heads can be reconstructed with only a small number of trainable parameters.

As a control, we train the same 5.24M Q/K parameters for one epoch without reinitialization. This setting reaches 72.56 Recall@GoldK on LongCite, compared with 68.75 when starting from random initialization. It also slightly outperforms the randomly initialized model trained for two epochs on LongCite (72.56 vs. 71.79). Nevertheless, it does not surpass the original untrained MV baseline of 73.52. One likely reason is that the relatively large learning rate of 2×10^{-3} used for this head-only reconstruction experiment can perturb an already useful pretrained solution.

Retrieval heads are generally viewed as specialized capabilities that emerge during pretraining (Zhang et al., 2025b; Fu et al., 2026; Lin et al., 2026). Our reconstruction experiment provides a complementary observation: once the rest of the pretrained representation space is fixed, task-specific Q/K matching can be relearned in a small set of attention heads with only 5.24M trainable parameters. This suggests that specialized attribution heads need not be treated solely as fixed structures inherited from pretraining; they can also be explicitly adapted, or even largely reconstructed, for downstream evidence localization.

D Limitations

Cross-dataset generalizability of attribution training. An attribution model trained on a single dataset can perform worse than the untrained model on other datasets, suggesting a potential overfitting issue in training. However, our experiments also show that training jointly on multiple datasets helps mitigate this overfitting, indicating that constructing more diverse attribution datasets is important for training.

Quantitative mechanistic evidence for value-vector cheating. The performance drop early in training, together with the growth in the maximum value norm, supports the shortcut explanation proposed in this paper, but we have not yet confirmed whether this norm inflation is concentrated on evidence tokens specifically, nor analyzed exactly how it affects the resulting ranking.