MAXSHAPLEY: TOWARDS FAIR CONTEXT ATTRIBUTION FOR GENERATIVE SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative search engines based on large language models (LLMs) are replacing traditional search, fundamentally changing how information providers are compensated. To sustain this ecosystem, we need fair mechanisms to attribute and compensate content providers based on their contributions to generated answers. We introduce MaxShapley, an efficient algorithm for fair credit attribution in generative search pipelines that retrieve external sources before generation. MaxShapley is a special case of the celebrated Shapley value; it leverages a decomposable max-sum utility function to compute attributions with polynomial-time computation in the number of documents, as opposed to the exponential worst-case cost of Shapley values. We evaluate MaxShapley on three multi-hop QA datasets (HotPotQA, MuSiQUE, MS MARCO); MaxShapley achieves comparable attribution quality to an LLM-based full Shapley computation, while consuming a fraction of its tokens—for instance, it gives up to a 9x reduction in resource consumption over prior state-of-the-art methods at the same attribution accuracy. We release open-source code and recalibrated datasets. A demonstration is available at `https://fair-search.com`.

1 INTRODUCTION

Generative search engines based on large language models are rapidly replacing traditional search (Gholami et al., 2026). Generative search engines like Perplexity AI (Perplexity AI, 2022) and Google Gemini (DeepMind, 2023) often follow a retrieval-augmented generation (RAG) pattern (Lewis et al., 2021; Guu et al., 2020; Izacard et al., 2022): they first retrieve relevant documents from a large corpus, then generate a concise response shown directly to the user.

Despite its convenience, generative search *fundamentally disrupts incentive structures for content providers*. Traditional search directed users to source websites, where providers monetized visits through advertising (Zeff and Aronson, 1999). Generative search instead answers directly in the interface, reducing visits to original sources. Traffic losses are already visible for many content providers (Athena Chapekis, 2025; Ritchie, 2025): Seer Interactive estimates that, as of September 2025, click-through rates to organic content are 62% lower when Google's AI Overviews are present than when they are not (McDonald, 2025). According to recent reports (Economist, 2025; SimilarWeb, 2025), the fraction of worldwide web traffic produced by traditional search fell about 5% from June 2024 to June 2025, with some sources estimating a drop of up to 25% (Sommerfeld et al., 2025). Some media organizations call this decline an "extinction-level event" (Allyn, 2025). At the same time, source citations in generative search, typically produced through prompt engineering, have been shown to be highly unreliable (Strauss et al., 2025) and easily manipulated (Nestaas et al., 2025).

Content providers are starting to push back with lawsuits against generative search providers for reduced traffic and lost revenue (Novet and Elias, 2025; Flynn, 2025; Perez, 2023). Complementary lawsuits target AI companies' use of copyrighted training material (The New York Times Company v. Microsoft Corporation et al., 2023; Bartz v. Anthropic PBC, No. 69058235, 2024), resulting in billions of dollars in liabilities and growing distrust from content providers (Press, 2025).

Industry responses are emerging: some generative search engines promise provider compensation (Gist, 2026; Gunasekara et al., 2024), and infrastructure providers now let sites block AI crawlers or charge per crawl (Allen and Netwon, 2025). Yet the compensation rules remain opaque, with no clear evidence that compensation is tied to a source's *relevance* to the generated answer.

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.

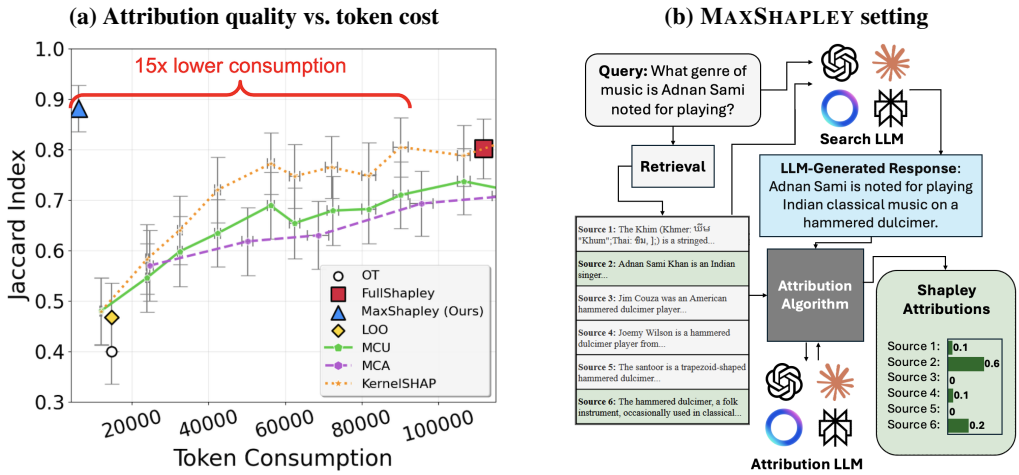

Figure 1: **Left:** On MuSiQUE with GPT-4.1 nano, MaxShapley reaches Jaccard 0.88 (↑) using 5.4% of FullShapley's tokens; KernelSHAP needs 15× MaxShapley's tokens for similar quality. **Right:** MaxShapley acts in a black-box setting, parallel to a search LLM. It takes as input a query q, an answer a, and source documents S, and outputs attribution scores for each document.

Problem statement and status quo. Long-term, the business model for generative search engines will need to evolve to compensate content providers for their contributions (Khosrowi et al., 2024). Early academic efforts to rethink the LLM ads ecosystem have primarily focused on sponsored search auctions for LLMs (Criddle, 2024; Bergemann et al., 2024; Hajiaghayi et al., 2024; Duetting et al., 2024; Banchio et al., 2024; Feizi et al., 2023), which do not directly benefit organic content providers. In this paper, we tackle the algorithmic *attribution* piece of this puzzle: our goal is to define a method for attributing generative search results to specific original sources, so that content providers can be fairly compensated for their work. In particular, we define fairness according to common axiomatic properties (Section 2). A key operational requirement is that our algorithm should be practical for existing generative search pipelines by minimizing the number and size of queries to an LLM oracle.

In the broader machine learning community, variants of the attribution problem have been used to interpret and explain the behaviors of complex machine learning models, which is crucial for building trustworthy and reliable AI systems. Notable high-impact works include Datamodels (Ilyas et al., 2022), TRAK (Park et al., 2023) and Data Shapley (Ghorbani and Zou, 2019; Wang et al., 2025a; 2024) for training-time attribution to training samples, and LIME (Ribeiro et al., 2016) and KernelSHAP (Lundberg and Lee, 2017) for inference-time attribution between inputs and features. In contrast, our work addresses inference-time credit attribution.

Another relevant line of work is *context attribution for retrieval-augmented LLMs*, which aims to identify which retrieved context spans support or explain the final answer generated by an LLM (Cohen-Wang et al., 2024; 2025; Qi et al., 2024; Chuang et al., 2025; Ding et al., 2024; Liu et al., 2025a; Xiao et al., 2025; Hirsch et al., 2025). However, most existing context attribution methods focus on *fine-grained explainability*, debugging, or citation, rather than *fair and quantitative* credit attribution to information providers from an economic perspective. A few recent works are closer to our setting and explore the use of Shapley value (Shapley et al., 1953) for attribution to information sources (Xiao et al., 2025; Nematov et al., 2025; Wang et al., 2025b), where Nematov et al. (2025) found that the KernelSHAP method (Lundberg and Lee, 2017) outperforms other Shapley-based baselines in terms of both attribution accuracy and computational efficiency. However, Shapley-based attribution typically requires repetitive trials to evaluate the contributions of sources, a known limitation of Shapley values (Nematov et al., 2025; Wang et al., 2025b), making prior work computationally infeasible in latency-sensitive search settings (Figure 1, Section 4).

1.1 Our Contribution

We study fair attribution of information sources for generative search. We propose MaxShapley, a Shapley-value-based attribution algorithm that quantifies sources' contributions to the final answer by treating providers as players in a cooperative game.

MAXSHAPLEY operates downstream of the generative search pipeline: it only takes the query, retrieved sources, and generated response as input (see Figure 1, right). This design has two practical advantages. First, **MAXSHAPLEY requires only black-box access**: it does not need token-probability or log-likelihood access to the search LLM, which is required by ContextCite-style and log-likelihood-based RAG attribution utilities (Cohen-Wang et al., 2024; Nematov et al., 2025). It also does not assume reference answers and remains customizable to different evaluation criteria, such as relevance, accuracy, and helpfulness. Second, **MAXSHAPLEY is computationally efficient because it avoids repeated context ablations**. Existing utility-based context attribution methods remove or sample subsets of input sources, and then evaluate the impact of the ablation to derive the marginal contribution of each source (Cohen-Wang et al., 2024; Lundberg and Lee, 2017; Nematov et al., 2025). This ablation loop is the dominant cost because each sampled subset requires another model or utility evaluation. In contrast, MAXSHAPLEY scores source-keypoint support once and computes the Shapley values exactly using a *decomposable max-sum utility function*. After scoring, the attribution step is deterministic arithmetic with no subset sampling; the sorted-scan algorithm of Littlechild and Owen (1973) computes the exact Shapley values for each max-function subproblem in $O(m \log m)$ time, improving over the $O(m^2)$ routine of Lundberg and Lee (2017).

Empirically, MAXSHAPLEY preserves attribution quality while substantially reducing token cost. It achieves strong correlation with brute-force Shapley values computed from an aggregate LLM-as-a-judge utility function (Kendall-$\tau > 0.5$) and strong alignment with human annotations (Jaccard index > 0.85), using less than 6% of the token cost (Figure 3). Compared to KernelSHAP, our strongest Shapley approximation baseline, **MAXSHAPLEY reaches the same alignment with human annotations (Jaccard) at less than 7% the cost** (Figure 1 , Figure 3). Our contributions are:

1. **Algorithmic:** We propose MAXSHAPLEY (Section 3), a novel and efficient Shapley-value-based algorithm for fair attribution of information sources in generative search, including an exact $O(m \log m)$ algorithm for the max-function Shapley subproblem underlying our construction. We show that MAXSHAPLEY can be viewed as a special case of a more general class of coverage games, which present path towards corroboration-aware attribution (Appendix C).
2. **Empirical:** We demonstrate through extensive evaluations (Section 4) that MAXSHAPLEY achieves a significantly better tradeoff between attribution quality and efficiency than existing methods. These gains are robust to choices of attribution LLM and other hyperparameters.
3. **Artifacts:** We release open-source code, manually re-annotated subsets of HotPotQA, MuSiQUE, and TREC[1], and a demonstration[2]

2 PROBLEM SETUP AND PRELIMINARIES

For a user query q, the system retrieves an ordered list of sources $S = (s_1, \ldots, s_m)$, where each source is a document, passage, or snippet. The search LLM Ψ then generates the answer $a = \Psi(q, S)$. The attribution task is to assign a score ϕ_i to each source s_i, quantifying its contribution to the displayed answer a. These scores can be used for source ranking, auditing, or credit allocation.

We use index sets to denote source sets. Let $[m] = \{1, \ldots, m\}$. For any subset $I \subseteq [m]$, write $S_I = \{s_i : i \in I\}$ for the corresponding subset of sources. The attribution module has access to the query q, the answer a, and black-box access to an attribution LLM Ψ_A, which may be different from the search LLM Ψ. In evaluation only, some datasets provide a ground truth reference answer $\tilde{a}$ (distinct from the LLM response a) and binary relevance labels $\tilde{y}_i \in \{0, 1\}$ indicating whether source s_i is annotated as relevant. MAXSHAPLEY does not require labels or reference answers.

Cooperative game and Shapley attribution. A cooperative game consists of a set of players and a utility function that assigns a value to every subset of players. In our attribution problem, the players are the retrieved sources, indexed by $[m]$. We represent a game by a subset utility $U : 2^{[m]} \to \mathbb{R}$. The value $U(I)$ represents the utility obtained when only the sources in I are available. We shift utilities so that $U(\emptyset) = 0$; subtracting a constant from all subset values does not change any marginal contribution. An attribution rule maps each utility function U to a vector $\boldsymbol{\phi}^U = (\phi_1^U, \ldots, \phi_m^U)$. We use the standard Shapley desiderata (Shapley et al., 1953):

[1] https://github.com/anonymousresearcher298/FairAttribution

[2] https://fair-search.com

1. **Efficiency**: $\sum_{i=1}^{m} \phi_i^U = U([m])$.
2. **Symmetry**: if $U(I \cup \{i\}) = U(I \cup \{j\})$ for every $I \subseteq [m] \setminus \{i, j\}$, then $\phi_i^U = \phi_j^U$.
3. **Null player**: if $U(I \cup \{i\}) = U(I)$ for every $I \subseteq [m] \setminus \{i\}$, then $\phi_i^U = 0$.
4. **Additivity**: for utilities U_1, U_2, the attribution for $U_1 + U_2$ satisfies $\phi_i^{U_1+U_2} = \phi_i^{U_1} + \phi_i^{U_2}$.

The Shapley value is the unique attribution rule satisfying these four axioms. Equivalently, it is the expected marginal contribution of a source when sources arrive in a uniformly random order:

$$\phi_i^U = \mathbb{E}_{\pi \sim \mathrm{Unif}(\Pi_m)} \left[U(P_i^\pi \cup \{i\}) - U(P_i^\pi) \right], \tag{1}$$

where Π_m is the set of permutations of $[m]$, and P_i^π is the set of indices appearing before i in permutation π. Shapley value is notorious for its computational intractability: without special structure in the utility U, exact Shapley computation requires enumeration over all 2^m subsets.

2.1 Baseline attribution methods

FullShapley. A natural baseline computes the exact Shapley value in Equation 1 for a black-box utility function obtained by re-running the full search-generation pipeline on each subset $I \subseteq [m]$ of retrieved sources and measuring answer quality as follows:

$$a_I = \Psi(q, S_I) \qquad \textit{answer generation from a source subset } I \tag{2}$$

$$\widetilde{U}(I) = \textsc{Judge}_{\Psi_A}(q, a_I, S_I; p) \qquad \textit{answer-quality grading} \tag{3}$$

$$U(I) = \widetilde{U}(I) - \widetilde{U}(\emptyset) \qquad \textit{marginal gain over the null case} \tag{4}$$

Here p denotes the judging prompt and scoring rule, which may encode correctness, relevance, completeness, and, for oracle baselines, optional access to a reference answer. Following the LLM-as-a-judge paradigm (Zheng et al., 2023; Liu et al., 2023b), we implement $\textsc{Judge}_{\Psi_A}$ by prompting the attribution LLM Ψ_A to score the generated answer under these criteria (prompt in Appendix E.2). This baseline has cost exponential in the number of retrieved sources. In our evaluation, FullShapley is also given access to reference answers when available, making it an oracle-style baseline.

Leave-One-Out (LOO) and Optimal Transport (OT). LOO attribution (Koh and Liang, 2017; Liu et al., 2025a) assigns

$$\phi_i^{\mathsf{LOO}} = U([m]) - U([m] \setminus \{i\}).$$

It is cheap and intuitive, but not budget balanced: the scores need not sum to $U([m])$. It also under-credits redundant support; if two sources provide the same evidence, removing either one may barely change the utility. We additionally evaluate a sentence-level LOO variant in Appendix G. For OT, we adapt GiLOT (Li et al., 2024) by treating each source document as the perturbation unit and measuring influence through optimal-transport distances between generations with(out) that source.

Monte-Carlo Shapley (MCU, MCA). Monte-Carlo methods approximate Equation 1 by sampling permutations and averaging observed marginal contributions (Michalak et al., 2013; Mitchell et al., 2022). **MCU** samples permutations uniformly. **MCA** pairs each sampled permutation with its reverse to reduce variance. These methods approach the Shapley value with enough samples, but in LLM settings each sampled subset still corresponds to an expensive utility evaluation.

KernelSHAP. KernelSHAP (Lundberg and Lee, 2017) estimates Shapley values by fitting a weighted linear surrogate model over sampled subsets. It is a strong general-purpose approximation, but its quality depends on the number and distribution of sampled subsets; empirically, it requires far more LLM calls than MaxShapley to approach FullShapley-quality attribution (Section 4).

3 MaxShapley: Efficient, Fair Attribution in Generative Search

We present MaxShapley, a Shapley-based algorithm for fair attribution in generative search.

Our main design is a novel utility function for generative search and an efficient algorithm for its Shapley values. Figure 2 illustrates the pipeline. We observe that retrieved information sources in generative search provide both complementary and overlapping information, and attribution should account for both:

- **Cooperation in providing complementary information.** Sources contribute distinct information to the final answer (e.g., articles covering different market sectors for a stock trend query), forming a *cooperative* game.
- **Competition in providing overlapping information.** When sources cover overlapping perspectives of the same information, the more relevant source should receive more credit, showing a *competitive* nature even in the cooperative game.

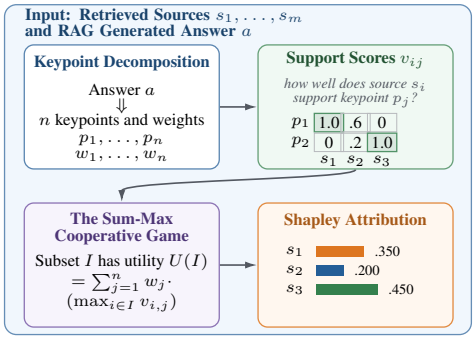

Figure 2: MAXSHAPLEY pipeline: answer decomposition, source-keypoint support scoring, max-sum utility, and exact Shapley attribution.

We define the utility in two steps. First, the attribution LLM Ψ_A decomposes the answer a into key points $P = \{p_1, \ldots, p_n\}$ (top left of Figure 2; prompt in Appendix E.2); examples are in Appendix E.3. Second, for each source-keypoint pair, Ψ_A assigns a support score $v_{i,j}$ measuring how well source s_i supports key point p_j (top right of Figure 2; prompt in Appendix E.2). The simple pairwise interface uses nm LLM calls; batched prompts can reduce API requests to m calls (one source against all key points) or n calls (one key point against all sources), with comparable token/FLOP cost up to batching and KV-cache constants. After that, all utility and Shapley computations are deterministic.

For a subset $I \subseteq [m]$, each key point is covered by its best supporting source in S_I, and the subset utility is the weighted sum of these best-support scores:

$$U_{\text{MAXSHAPLEY}}(I) = \sum_{j=1}^{n} w_j \max_{i \in I} v_{i,j}, \tag{5}$$

where $\max_{i \in \emptyset} v_{i,j} = 0$ (bottom left, Figure 2). Here, w_j is the weight of key point p_j. Given the support scores v_{ij}, evaluating $U_{\text{MAXSHAPLEY}}$ requires no further LLM calls.

This sum-max structure resembles the MaxSim score in ColBERT (Khattab and Zaharia, 2020; Santhanam et al., 2021), a state-of-the-art retrieval method (Santhanam et al., 2021), which further justifies our design. Our approach differs in two ways: (1) we compute scores at the *key-point* level via LLM-as-a-judge rather than at the token level, capturing holistic semantic information; (2) we connect this structure to fair attribution through Shapley values (Section 3.1). In fact, we also conducted experiments with an attribution mechanism based on Shapley values with the original MaxSim score, but the results showed that the token-level embeddings fail to distinguish between low-level linguistic similarity and deeper-level of semantic relevance.

3.1 EFFICIENT SHAPLEY VALUE COMPUTATION FOR THE NEW UTILITY FUNCTION

A key advantage of this utility function is that it admits exact Shapley value computation in polynomial time, avoiding Monte Carlo approximations entirely. First, note that the sum-max structure of Equation 5 decomposes into n independent maximization games, one per key point. Define

$$U^j_{\text{MAX}}(I) = \max_{i \in I} v_{i,j}, \tag{6}$$

with corresponding Shapley values $\phi_i^{U^j_{\text{MAX}}}$, for $j \in [n]$. The additivity axiom of Shapley values implies that $\phi_i^{U_{\text{MAXSHAPLEY}}} = \sum_{j=1}^{n} w_j \phi_i^{U^j_{\text{MAX}}}$. Hence, we only need the per-keypoint Shapley values $\phi_i^{U^j_{\text{MAX}}}$ based on the utility in (6).

Shapley value for key-point level maximization games. Each per-keypoint game $U^j_{\text{Max}}(I) = \max_{i \in I} v_{i,j}$ is a maximization game with a particularly simple exact Shapley computation due

Algorithm 1 MAXSHAPLEY attribution algorithm

INPUT: Query q, retrieved sources $S = \{s_1, \ldots, s_m\}$, generated answer a, attribution LLM Ψ_A.
OUTPUT: Attribution score ϕ_i for each source s_i.

Stage I: Generating support scores
Use Ψ_A to decompose answer a into key points $P = \{p_1, \ldots, p_n\}$ and weights $w_1, \ldots, w_n$.
for $j = 1$ to n **do**
  for $i = 1$ to m **do**
    Use Ψ_A to score how well source s_i supports key point p_j; denote the score by $v_{i,j}$.
  end for
end for
Stage II: Computing Shapley values for the utility function $U_{\text{MAXSHAPLEY}}$
Initialize $\phi_i \leftarrow 0$ for every source $i \in [m]$.
for $j = 1$ to n **do**
  $\{\phi_{i,j}^{\mathsf{Max}}\}_{i=1}^{m} \leftarrow$ MAXGAMESHAPLEY$(v_{1,j}, \ldots, v_{m,j})$.
  for $i = 1$ to m **do**
    $\phi_i \leftarrow \phi_i + w_j \phi_{i,j}^{\mathsf{Max}}$.
  end for
end for
return $\phi_1, \ldots, \phi_m$.

Subroutine **MAXGAMESHAPLEY**$(v_1, \ldots, v_m)$***: Shapley values for*** $U(I) = \max_{i \in I} v_i$
Sort players so that $0 \le v_1 \le v_2 \le \cdots \le v_m$, preserving original indices.
Set $v_0 \leftarrow 0$ and $c \leftarrow 0$.
for $r = 1$ to m **do**
  $c \leftarrow c + (v_r - v_{r-1})/(m - r + 1)$ // *distribute the marginal value to players* $\{r, r+1, \ldots, m\}$
  $\phi_r \leftarrow c$
end for
return $\{\phi_i\}_{i \in [m]}$, mapped back to the original source order.

to Littlechild and Owen (1973). By Corollary C.2, its exact Shapley values can be computed by sorting the m scores once and scanning the interval endpoints. The MAXGAMESHAPLEY subroutine in Algorithm 1 implements this corollary in $O(m \log m)$ time, compared to $O(m \cdot 2^m)$ for brute-force Shapley and the $O(m^2)$ Max SHAP routine described by Lundberg and Lee (2017). Appendix C formalizes this as a special case of a general information coverage game. Thus, once the source-keypoint scores are computed, no subset enumeration or quadratic pairwise scan is required.

The full MAXSHAPLEY is presented in Algorithm 1, with implementation details in Appendix H.

Proposition 3.1. Given source-keypoint support scores $\{v_{i,j}\}$ and key-point weights $\{w_j\}$, MAXSHAPLEY (Algorithm 1) returns the exact Shapley values $\phi_i^{U_{\text{MAXSHAPLEY}}}$ for the best-support utility $U_{\text{MAXSHAPLEY}}$ in Equation 5. Its computational complexity is $O((T + nmL)|\Psi| + nm \log m)$, where T upper bounds the token budget of the non-repeated MAXSHAPLEY prompts and generated outputs, L upper bounds each source-keypoint scoring prompt length, including both the prompt template and the source length, and $|\Psi|$ is the maximum per-token FLOP cost of the attribution LLM.

The proof follows from Shapley additivity and the max-game computation in Appendix C.2. Table 1 compares baselines' access assumptions, Shapley desiderata, and attribution-time inference cost under a normalized per-token compute model.

4 EMPIRICAL EVALUATION

4.1 EVALUATION SETUP

We evaluate MAXSHAPLEY's *quality of attribution* and *efficiency*. In this evaluation, we set MAXSHAPLEY's key-point weights uniformly, $w_j = 1/n$ for simplicity.

Baselines. We compare to the baselines introduced in Section 2, including **FullShapley**, **LOO**, **OT**, **MCU**, **MCA**, and **KernelSHAP**. Since we view FullShapley as our strongest baseline, we gave its **Judge** function access to a ground truth response $\tilde{a}$ (Prompt in Appendix E.2).

Quality Metrics. These metrics measure the agreement with *FullShapley* and/or groundtruth labels:

Method	Access	Shapley desiderata				Attribution-time cost
		Eff.	Sym.	Null	Add.	
Black-box methods						
FullShapley	Black box	✓	✓	✓	✓	$O(2^m(mL+T)\|\Psi\| + m2^m)$
LOO	Black box	✗	✓	✓	✓	$O(m(mL+T)\|\Psi\| + m)$
MCU/MCA	Black box	✓*	✓*	✓*	✓*	$O(km(mL+T)\|\Psi\| + km)$
MaxShapley (Ours)	**Black box**	✓	✓	✓	✓	$\mathbf{O((T+nmL)\|\Psi\| + nm\log m)}$
Model-internal / training-based methods						
AttriBot (Liu et al., 2025a)	Model/proxy	✗	✓	✓	✓	Target LOO: $O(m(mL+T)\|\Psi\|)$; proxy replaces $\|\Psi\|$ with $\|\Psi_{\text{proxy}}\|$
ARC-JSD (Li et al., 2025)	Logits	✗	✓	✓	✓	$O(m(mL+T)\|\Psi\| + mT\|\mathcal{V}\|)$
SelfCite (Chuang et al., 2025)	Logprobs/train†	N/A	N/A	N/A	N/A	Per statement: $O(N(mL+T)\|\Psi\|)$ plus optional SimPO training

Table 1: Comparison of attribution methods. m: sources; n: key points; L: source-level scoring context length (including the source length); T: generated-output budget; k: sampled permutations; N: SelfCite best-of-N candidates; $|\mathcal{V}|$: vocabulary size; $|\Psi|$: per-token model cost. *Satisfied in expectation. †Requires training.

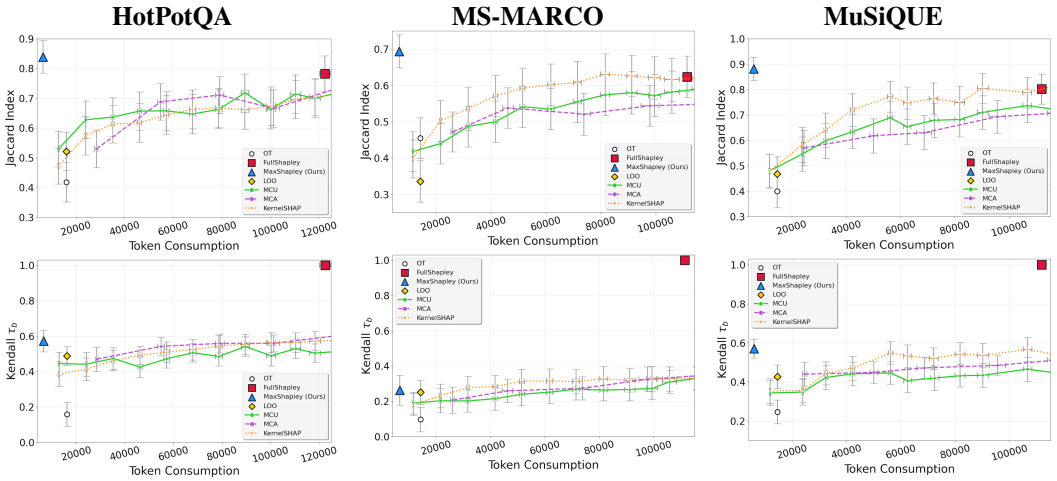

Figure 3: Attribution quality (Jaccard index w.r.t. ground truth (top), Kendall τ_b w.r.t. FullShapley (bottom)) versus token consumption for attribution algorithms on all three datasets, using GPT-4.1 nano. MaxShapley and KernelSHAP achieve comparable Jaccard index with FullShapley, which uses at least 12× more tokens. MaxShapley has a strong ordinal correlation (Kendall's τ_b) with FullShapley on HotPotQA, MuSiQUE. On MS-MARCO, MaxShapley has a moderate ordinal correlation. For similar correlations with FullShapley, KernelSHAP consumes 4-9× more tokens than MaxShapley.

- **Jaccard@K** between the ground truth relevance labels for each document (see **Datasets**) and the top-K elements of the MaxShapley vector. Let R be the ground truth relevant sources for a query (as annotated in a dataset) and let $K = |R|$. Let $T = \text{Top}_K(\hat{\phi})$. $\text{Jaccard@}K = \frac{|T \cap R|}{|T \cup R|}$. Jaccard@K $\in [0, 1]$ with 1.0 indicating perfect agreement between sets (higher is better). In Section 4.1.1, we show a precise connection between Jaccard@K and a method's payment allocation.
- **Kendall's τ_b (Kendall, 1938)** between the MaxShapley and FullShapley vectors. Ordinal agreement between rankings induced by $\hat{\phi}$ and $\phi^\star$; $\tau_b \in [-1, 1]$ with 0.0 indicating no ordinal correlation and 1.0 indicating perfect correlation (higher is better).

Datasets. We evaluate three multi-hop question answering datasets: **(1) HotpotQA** (Yang et al., 2018): Reasoning queries over Wikipedia documents. **(2) MuSiQUE** (Trivedi et al., 2022): Structured two-hop questions in full-Wiki setting. **(3) MS MARCO** (Bajaj et al., 2018; Craswell et al., 2020): Passage ranking with graded relevance judgments from the TREC 2019 Deep Learning Track.

Annotation. Because the original dataset labels can be noisy, we manually re-annotated focused subsets of 100 HotPotQA queries, 100 MuSiQUE queries, and 95 MS-MARCO queries, with six candidate sources per query and consensus labels from two annotators. Annotation protocol and

agreement statistics are in Appendix E.5. We include an ablation varying the number of sources in Appendix G to empirically compare scaling behavior between MAXSHAPLEY and KernelSHAP. All methods are evaluated on these annotated subsets across 100 independent runs; we report means and standard errors. We also evaluate MAXSHAPLEY on the original, larger datasets in Appendix G.

Evaluation Limitations. We treat both FullShapley and manually-annotated relevance as ground truth for attribution quality, although neither is perfect. LLM-as-a-judge exhibits scoring inconsistencies even at temperature 0 (Section 4.2), affecting all Shapley methods, including FullShapley. Additionally, manually-annotated relevance measures a related but distinct concept from Shapley attribution. Lacking a single ground truth, we measure association with both of these quantities.

4.1.1 ECONOMIC IMPLICATIONS OF JACCARD@K: RELEVANCE ALIGNMENT

In this section, we establish that a high Jaccard@K score implies that only a small fraction of the payment budget must be reassigned to restore a notion called *relevance-order consistency*. We denote by $B(\boldsymbol{\phi}) := \sum_{i=1}^{m} \phi_i$ the total budget and assume $B(\boldsymbol{\phi}) > 0$. Define the normalized allocation $\boldsymbol{p}(\boldsymbol{\phi}) := \boldsymbol{\phi}/B(\boldsymbol{\phi})$, so that $\sum_{i=1}^{m} p_i(\boldsymbol{\phi}) = 1$. We first derive a necessary and sufficient condition for exact relevance recovery.

Proposition 4.1. For any nonnegative payment allocation $\boldsymbol{\phi}$ with $B(\boldsymbol{\phi}) > 0$, there exists a top-K tie-breaking rule such that $\mathrm{Top}_K(\boldsymbol{\phi}) = R$, equivalently Jaccard@$K = 1$, if and only if $\boldsymbol{p}(\boldsymbol{\phi}) \in \mathcal{F}(R)$, where $\mathcal{F}(R) := \{\boldsymbol{p} \in \mathbb{R}^m_{\geq 0} : \sum_{i=1}^{m} p_i = 1,\ p_i \geq p_k,\ \ \forall i \in R,\ k \in [m] \setminus R\}$.

The set $\mathcal{F}(R)$ formalizes a natural relevance-order consistency requirement for a good payment allocation: each relevant source should receive at least as much credit as any irrelevant source. We next quantify deviations from relevance-order consistency as the minimum (normalized) payment that must be reassigned to restore relevance-order consistency.

Definition 4.2 (Relevance-consistency loss)**.** For a payment allocation $\boldsymbol{\phi}$ with $B(\boldsymbol{\phi}) > 0$, define

$$\mathcal{L}(\boldsymbol{\phi}) = \frac{1}{2} \min_{\boldsymbol{p}^0 \in \mathcal{F}(R)} \left\| \boldsymbol{p}(\boldsymbol{\phi}) - \boldsymbol{p}^0 \right\|_1 = \min_{\boldsymbol{p}^0 \in \mathcal{F}(R)} \sum_{i=1}^{m} \left(p_i(\boldsymbol{\phi}) - p_i^0 \right)_+ . \tag{7}$$

In the above, $(x)_+ := \max\{x, 0\}$. The factor $1/2$ is because the ℓ_1 distance counts each reassigned payment share twice: once when it is removed from one source and once when it is added to another. The equivalent positive-part form counts the total share removed from sources above their target shares. We next derive an upper bound on this loss under MaxShapley.

Theorem 4.3. *Let $V = (v_{ij})_{i \in [m],\, j \in [n]}$ denote the support-score matrix used by MaxShapley. Then, for any fixed top-K tie-breaking rule with J denoting the resulting Jaccard@K score, we have*

$$\mathcal{L}(\boldsymbol{\phi}^{\mathrm{MaxShapley}}) \leq K\, \Delta(V) \min\left\{ \frac{1 - J}{1 + J}, \frac{1}{K + 1} \right\}, \tag{8}$$

where $\Delta(V) = \dfrac{\sum_{j=1}^{n} w_j \left(\max_{k \in [m] \setminus R} v_{kj} - \min_{i \in R} v_{ij} \right)_+}{\sum_{j=1}^{n} w_j \max_{i \in [m]} v_{ij}}$, *with the convention* $\max_{k \in \varnothing} v_{kj} = 0$.

Here, $\Delta(V) \in [0, 1]$ is the normalized support-order defect, measuring positive gaps between irrelevant- and relevant-source support scores. The theorem establishes that a high Jaccard@K score implies a low cost of restoring relevance-order consistency. We defer all proofs in this section to Appendix D, and empirically evaluate the loss and its bound under MaxShapley in Appendix F.

4.2 MAIN RESULTS

(1) MAXSHAPLEY achieves a better tradeoff among baselines between attribution quality efficiency by a significant margin. Figures 3 show the tradeoff between token consumption and attribution quality, measured by Jaccard index with ground truth and Kendall's τ_b with FullShapley, using GPT-4.1 nano. MAXSHAPLEY consistently outperforms LOO, OT, MCU, MCA, and KernelSHAP across all datasets and metrics. On MuSiQUE, KernelSHAP uses $\geq 15\times$ more tokens than MAXSHAPLEY to reach similar Jaccard indices, while Monte Carlo methods require 15-20$\times$ more.

For rank correlation measured by Kendall's τ_b (Figure 3, bottom), MaxShapley achieves a strong ordinal correlation with FullShapley on MuSiQUE and HotPotQA, while KernelSHAP requires 7-9$\times$

more tokens to reach the same correlation quality. On MS MARCO, MaxShapley achieves a moderate correlation with KernelSHAP achieving the same with 4× more tokens.[3]

On MS-MARCO, we observe a degradation in the quality of attribution across all Shapley attribution methods. MS-MARCO, is a less curated dataset than HotPotQA and MuSiQUE, with sometimes confusing information source content (even for humans). As such, the Search LLM had more trouble forming coherent and correct responses to queries given information sources.

Note that MAXSHAPLEY has a higher Jaccard index with the ground truth than FullShapley on HotPotQA and MS-MARCO (Figure 3). This is because semantically-similar answers can receive different JUDGE_{Ψ_A} scores, which particularly affects FullShapley-style baselines.

We provide the evaluation results on the full, original MuSiQUE dataset in Appendix G.

(2) Localized LLM judging stabilizes Shapley attribution. The black-box utility in Equation 4 re-runs generation for each source subset, and asks JUDGE_{Ψ_A} to score the whole answer. This couples Shapley attribution to repeated LLM judgments. As shown in Table 2, two answers that are semantically equivalent to human readers receive mean JUDGE_{Ψ_A} scores of 0.3 and 1.0 over 10 deterministic runs, while identical inputs show little variance. This pattern matches prior findings that LLM evaluations are sensitive to wording, source ordering, and context length (Salinas and Morstatter, 2024; Lin et al., 2025).

MAXSHAPLEY instead uses the attribution LLM more narrowly: decomposing the displayed answer into key points and scoring each source-keypoint pair's relevance. This design avoids the repeated, complex judgments required by FullShapley. MAXSHAPLEY's high Jaccard scores (Figure 3) suggest that its narrower LLM-as-a-judge use preserves semantic signal and reduces instability.

4.3 ABLATIONS

We conducted an ablation study on MAXSHAPLEY and baselines. See more details in Appendix 4.3.

Model Selection. For our attribution LLM, we evaluated GPT-4.1 nano (OpenAI), Claude Haiku 4.5, Claude 3.5, and Claude Sonnet 4 (Anthropic), and conducted our main experiments using only the first two. Haiku 4.5 achieved notably higher attribution quality than GPT-4.1 nano with moderately increased token consumption and cost, but with a noticeable increase in execution time, excluded Sonnet 4 due to prompt incompatibility and higher cost and Claude Haiku 3.5, which was retired prior to our dataset expansion and replaced by Haiku 4.5 (see Figure 10 in Appendix G).

Clipping. Even at temperature 0, some baselines assign tiny nonzero attribution scores to irrelevant sources, destabilizing their rank ordering. We therefore clip all attributions below 0.05 and renormalize for every baseline *except* MAXSHAPLEY, whose max operation does not spread small scores across unsupported sources. We show the effect of clipping on FullShapley in Appendix G.

Positional Bias. LLMs can over-attend to information near context boundaries (Liu et al., 2023a), so source order can affect attribution. On 30 MuSiQUE samples with exactly two relevant sources, Haiku 3.5 gave MAXSHAPLEY a 0.12 higher Jaccard score when the relevant sources were placed first than when all sources were randomly shuffled (Appendix E.5 and G). We therefore randomly shuffle source order before each LLM call in our evaluation, preventing systematic ordering bias.

5 CONCLUSION, LIMITATIONS, AND FUTURE DIRECTIONS

We present MAXSHAPLEY, a black-box Shapley-based algorithm for fair attribution to information sources in generative search. Our max-sum utility formulation preserves the Shapley fairness desiderata with an efficient runtime. Empirically, MAXSHAPLEY achieves strong agreement with FullShapley and human relevance annotations while using a small fraction of FullShapley's token cost. Several limitations and assumptions motivate future work. First, we operate under a black-box access assumption, motivated by a desire for third-party attribution. White-box attribution may also be desirable for some LLM providers. Another natural extension is a corroboration-aware utility that assigns additional value when multiple sources mutually support the same key point.

[3]There is no standard way to interpret the quality of a τ_b correlation, we follow Wicklin (2023), using ≥ 0.71 for very strong correlation, $>= 0.49$ for strong, $>= 0.26$ for moderate, and < 0.26 for weak/negligible.

6 AI USE STATEMENT

We used AI to implement code for experiments, which we checked manually. We also used AI to search for related work, though we determined and described the relevance of all cited papers ourselves. AI was used to draft and polish text, though all text was read and edited by one or more humans. We used AI to help develop the coverage game formulation (with input from the authors), and to draft initial description of this formulation, as well as to draft some of the proofs. AI was not used to formulate mathematical claims or interpret results. This paper did not require us to generate synthetic data, translate text, clean and reformat datasets, or do qualitative or thematic analysis. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

7 REPRODUCIBILITY

Our code and annotated datasets are available open-source to enable full reproducibility. We have uploaded them as supplementary material to the following anonymous repository.

REFERENCES

W. Allen and S. Netwon. Introducing pay per crawl: Enabling content owners to charge AI crawlers for access. https://blog.cloudflare.com/introducing-pay-per-crawl/, 7 2025. The Cloudflare Blog. [Online; accessed 2025-10-17].

B. Allyn. Will Google's AI Overviews kill news sites as we know them? *NPR Technology*, July 2025.

Anthropic. Claude 3.5 Haiku. https://www.anthropic.com/claude/haiku, 2024. Accessed: 2025-10-07.

Anthropic. Introducing Claude 4. https://www.anthropic.com/news/claude-4, 2025. Accessed: 2025-10-07.

A. L. Athena Chapekis. Google users are less likely to click on links when an ai summary appears in the results, 2025. URL https://www.pewresearch.org/short-reads/2025/07/22/google-users-are-less-likely-to-click-on-links-when-an-ai-summary-appears-in-the-results/.

P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary, and T. Wang. Ms marco: A human generated machine reading comprehension dataset, 2018. URL https://arxiv.org/abs/1611.09268.

M. Banchio, A. Mehta, and A. Perlroth. Ads in conversations. *arXiv preprint arXiv:2403.11022*, 2024.

Bartz v. Anthropic PBC, No. 69058235. U.S. District Court, Central District of California, 2024. URL https://www.courtlistener.com/docket/69058235/bartz-v-anthropic-pbc/.

D. Bergemann, M. Bojko, P. Dütting, R. P. Leme, H. Xu, and S. Zuo. Data-driven mechanism design: Jointly eliciting preferences and information. *arXiv preprint arXiv:2412.16132*, 2024.

Y.-S. Chuang, B. Cohen-Wang, Z. Shen, Z. Wu, H. Xu, X. V. Lin, J. R. Glass, S.-W. Li, and W. tau Yih. Selfcite: Self-supervised alignment for context attribution in large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=rKi8eyJBoB.

B. Cohen-Wang, H. Shah, K. Georgiev, and A. Madry. Contextcite: Attributing model generation to context. *Advances in Neural Information Processing Systems*, 37:95764–95807, 2024.

B. Cohen-Wang, Y.-S. Chuang, and A. Madry. Learning to attribute with attention, 2025. URL https://arxiv.org/abs/2504.13752.

N. Craswell, B. Mitra, E. Yilmaz, D. Campos, and E. M. Voorhees. Overview of the trec 2019 deep learning track, 2020. URL https://arxiv.org/abs/2003.07820.

C. Criddle. Perplexity in talks with top brands on ads model as it challenges google. https://www.ft.com/content/ecf299f4-e0a9-468b-af06-8a94e5f0b1f4, 9 2024. Online; accessed 2025-10-16.

G. DeepMind. Google gemini: A multimodal ai model. Blog post / technical announcement, 2023. URL https://blog.google/technology/ai/gemini-collection.

Q. Ding, L. Luo, Y. Cao, and P. Luo. Attention with dependency parsing augmentation for fine-grained attribution. *arXiv preprint arXiv:2412.11404*, 2024.

A. Dubey, Z. Feng, R. Kidambi, A. Mehta, and D. Wang. Auctions with llm summaries. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 713–722. Association for Computing Machinery, 2024. ISBN 9798400704901. doi: 10.1145/3637528.3672022. URL https://doi.org/10.1145/3637528.3672022.

P. Duetting, V. Mirrokni, R. Paes Leme, H. Xu, and S. Zuo. Mechanism design for large language models. In *Proceedings of the ACM Web Conference 2024*, pages 144–155, 2024.

T. Economist. Ai is killing the web. can anything save it?, 2025. URL https://www.economist.com/business/2025/07/14/ai-is-killing-the-web-can-anything-save-it.

S. Feizi, M. Hajiaghayi, K. Rezaei, and S. Shin. Online advertisements with llms: Opportunities and challenges. *arXiv preprint arXiv:2311.07601*, 2023.

K. Flynn. Penske Media sues Google over AI summaries taking traffic. *Axios*, 9 2025. URL https://www.axios.com/2025/09/14/penske-media-sues-google-ai. Online; accessed 2025-10-18.

S. Gholami, C. Firullo, C. Cheyre, and A. Acquisti. Beyond search: Llm adoption and web traffic concentration. *Available at SSRN 6238578*, 2026.

A. Ghorbani and J. Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pages 2242–2251. PMLR, 2019.

Gist. Gist: AI monetization solutions. https://gist.ai/, 2026. Online; accessed 2025-10-17.

A. Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, S. Wang, K. Zhang, Y. Wang, W. Gao, L. Ni, and J. Guo. A survey on llm-as-a-judge, 2025. URL https://arxiv.org/abs/2411.15594.

L. Gunasekara, A. Hsieh, L. Le, and J. Baron. The New O'Reilly Answers: The R in "RAG" Stands for "Royalties". https://www.oreilly.com/radar/the-new-oreilly-answers-the-r-in-rag-stands-for-royalties/, 6 2024. Online; accessed 2025-10-17.

K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang. Realm: Retrieval-augmented language model pre-training, 2020. URL https://arxiv.org/abs/2002.08909.

M. Hajiaghayi, S. Lahaie, K. Rezaei, and S. Shin. Ad auctions for llms via retrieval augmented generation. *Advances in Neural Information Processing Systems*, 37:18445–18480, 2024.

K. Harris, S. Prasad, and A. Trockman. In-context credit assignment via the core. *arXiv preprint arXiv:2605.06920*, 2026.

E. Hirsch, A. Slobodkin, D. Wan, E. Stengel-Eskin, M. Bansal, and I. Dagan. Laquer: Localized attribution queries in content-grounded generation. *arXiv preprint arXiv:2506.01187*, 2025.

A. Ilyas, S. M. Park, L. Engstrom, G. Leclerc, and A. Madry. Datamodels: Understanding predictions with data and data with predictions. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9525–9587. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ilyas22a.html.

G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave. Atlas: Few-shot learning with retrieval augmented language models, 2022. URL https://arxiv.org/abs/2208.03299.

M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938. ISSN 00063444. URL http://www.jstor.org/stable/2332226.

O. Khattab and M. Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020.

D. Khosrowi, F. Finn, and E. Clark. Engaging the many-hands problem of generative-ai outputs: A framework for attributing credit. *AI and Ethics*, 2024.

P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

J. Lee, D. Kwon, and K. Jin. Grade: Generating multi-hop qa and fine-grained difficulty matrix for rag evaluation, 2025. URL https://arxiv.org/abs/2508.16994.

P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL https://arxiv.org/abs/2005.11401.

R. Li, C. Chen, Y. Hu, Y. Gao, X. Wang, and E. Yilmaz. Attributing response to context: A jensen-shannon divergence driven mechanistic study of context attribution in retrieval-augmented generation. *arXiv preprint arXiv:2505.16415*, 2025.

X. Li, J. Chen, Y. Chai, and H. Xiong. GiLOT: Interpreting generative language models via optimal transport. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=qKL25sGjxL.

W. Lin, A. Gerchanovsky, O. Akgul, L. Bauer, M. Fredrikson, and Z. Wang. Llm whisperer: An inconspicuous attack to bias llm responses, 2025. URL https://arxiv.org/abs/2406.04755.

S. C. Littlechild and G. Owen. A simple expression for the shapley value in a special case. *Management science*, 20(3):370–372, 1973.

F. Liu, N. Kandpal, and C. Raffel. Attribot: A bag of tricks for efficiently approximating leave-one-out context attribution. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=9kJperA2a4.

N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the middle: How language models use long contexts, 2023a. URL https://arxiv.org/abs/2307.03172.

T. Liu, Z. Wang, M. Qin, Z. Lu, X. Chen, Y. Yang, and P. Shu. Real-time ad retrieval via llm-generative commercial intention for sponsored search advertising. *arXiv preprint arXiv:2504.01304*, 2025b.

Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment, 2023b. URL https://arxiv.org/abs/2303.16634.

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

T. McDonald. Google AI Overview Study - SEO & PPC CTR impact. *Seer Interactive*, 2 2025. URL https://www.seerinteractive.com/insights/ctr-aio. [Online; accessed 2025-12-17].

T. P. Michalak, K. V. Aadithya, P. L. Szczepanski, B. Ravindran, and N. R. Jennings. Efficient computation of the shapley value for game-theoretic network centrality. *Journal of Artificial Intelligence Research*, 46:607–650, 2013.

R. Mitchell, J. Cooper, E. Frank, and G. Holmes. Sampling permutations for shapley value estimation. *Journal of Machine Learning Research*, 23(43):1–46, 2022.

T. Mordo, M. Tennenholtz, and O. Kurland. Sponsored question answering. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 167–173, 2024.

I. Nematov, T. Kalai, E. Kuzmenko, G. Fugagnoli, D. Sacharidis, K. Hose, and T. Sagi. Source attribution in retrieval-augmented generation. *arXiv preprint arXiv:2507.04480*, 2025.

F. Nestaas, E. Debenedetti, and F. Tramèr. Adversarial search engine optimization for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=hkdqxN3c7t.

J. Novet and J. Elias. Chegg sues Google for hurting traffic as it considers alternatives. CNBC, 2 2025. URL https://www.cnbc.com/2025/02/24/chegg-sues-google-for-hurting-traffic-as-it-considers-alternatives.html. [Online; accessed 2025-10-18].

OpenAI. Introducing GPT-4.1 in the API. https://openai.com/index/gpt-4-1/, 2025. Accessed: 2025-10-07.

S. M. Park, K. Georgiev, A. Ilyas, G. Leclerc, and A. Madry. TRAK: Attributing model behavior at scale. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 27074–27113. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/park23c.html.

S. Perez. News publisher files class action antitrust suit against Google, citing AI's harms to their bottom line. *TechCrunch*, 12 2023. URL https://techcrunch.com/2023/12/15/news-publisher-files-class-action-antitrust-suit-against-google-citing-ais-harms-to-their-bottom-line/. Online; accessed 2025-10-18.

I. Perplexity AI. Perplexity ai: Answer engine. Website / Service, 2022. URL https://perplexity.ai.

T. A. Press. Anthropic to pay $1.5 billion to settle authors' copyright lawsuit, 2025. URL https://www.cbsnews.com/news/anthropic-copyright-lawsuit-class-action-settlement-authors-1-5-billion/.

J. Qi, G. Sarti, R. Fernández, and A. Bisazza. Model internals-based answer attribution for trustworthy retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, page 6037–6053. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.347. URL http://dx.doi.org/10.18653/v1/2024.emnlp-main.347.

M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

T. Ritchie. Ai overviews: How are publishers adapting to the rise of clickless search?, 2025. URL https://www.thecurrent.com/marketing-strategy-ai-overviews-publishers-rise-clickless-search.

A. Salinas and F. Morstatter. The butterfly effect of altering prompts: How small changes and jailbreaks affect large language model performance, 2024. URL https://arxiv.org/abs/2401.03729.

K. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, and M. Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488*, 2021.

L. S. Shapley. A value for n-person games. In *Contributions to the theory of games*, volume 2, pages 307–317. Princeton University Press, 1953. URL https://www.rand.org/pubs/research_memoranda/RM0670.html.

L. S. Shapley et al. *A value for n-person games*. Princeton University Press Princeton, 1953.

SimilarWeb, 2025. URL https://www.similarweb.com/.

N. Sommerfeld, M. McCurry, and D. Harrington. Goodbye Clicks, Hello AI: Zero-Click Search Redefines Marketing. *Bain & Company*, 2 2025. URL https://www.bain.com/insights/goodbye-clicks-hello-ai-zero-click-search-redefines-marketing/. Online; accessed 2025-12-04.

I. Strauss, J. Yang, T. O'Reilly, S. Rosenblat, and I. Moure. The attribution crisis in llm search results. *arXiv preprint arXiv:2508.00838*, 2025.

The New York Times Company v. Microsoft Corporation et al. No. 1:23-cv-11195, U.S. District Court, Southern District of New York, 2023. URL https://dockets.justia.com/docket/new-york/nysdce/1:2023cv11195/612697.

H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. Musique: Multihop questions via single-hop question composition, 2022. URL https://arxiv.org/abs/2108.00573.

J. T. Wang, Z. Deng, H. Chiba-Okabe, B. Barak, and W. J. Su. An economic solution to copyright challenges of generative ai. *arXiv preprint arXiv:2404.13964*, 2024.

J. T. Wang, P. Mittal, D. Song, and R. Jia. Data shapley in one training run. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Y. Wang, W. Zou, R. Geng, and J. Jia. Tracllm: A generic framework for attributing long context llms, 2025b. URL https://arxiv.org/abs/2506.04202.

R. Wicklin. How to interpret spearman and Kendall correlation coefficients. The DO Loop Blog, SAS Institute, Apr. 2023. URL https://blogs.sas.com/content/iml/2023/04/05/interpret-spearman-kendall-corr.html. Accessed: September 26, 2026.

Y. Xiao, Y. Zhu, S. Samyoun, W. Zhang, J. T. Wang, and J. Du. Tokenshapley: Token level context attribution with shapley value, 2025. URL https://arxiv.org/abs/2507.05261.

Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL https://arxiv.org/abs/1809.09600.

R. L. Zeff and B. Aronson. *Advertising on the Internet*. John Wiley & Sons, Inc., 1999.

L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685.

APPENDIX

A ADDITIONAL RELATED WORK

LLMs and Online Advertisement. LLMs have been widely used in online advertisement systems due to their strong natural language understanding and generation capabilities (Feizi et al., 2023; Liu et al., 2025b; Duetting et al., 2024). Many prior works have explored mechanism design and auction design for LLM-based advertisement systems (Hajiaghayi et al., 2024; Duetting et al., 2024; Banchio et al., 2024; Mordo et al., 2024; Dubey et al., 2024). The setting, however, is *orthogonal* to our work, where the focus is on the interaction between advertisers and the platform that serves the ads, where the advertisers are typically bidding for user attention. Our setting focuses on the interaction between information providers and generative search providers, where the information providers passively provide information to the service providers and usually only display ads from a third-party advertisement platform. In fact, the two settings can be considered complementary, where the fair attribution scores from MAXSHAPLEY can be used as a passive "bid" for the information providers to participate in the auction-based advertisement systems.

Attribution in Machine Learning. The attribution problem has also been extensively studied in the broader machine learning community. For training-time attribution, Datamodels (Ilyas et al., 2022) and TRAK (Park et al., 2023) work by learning a prediction model on the impact of each training data point on the model's performance, while Data Shapley (Ghorbani and Zou, 2019; Wang et al., 2025a; 2024) applies the Shapley value concept to quantify the contribution of each training data point to the model's performance. For inference-time attribution, LIME (Ribeiro et al., 2016) works by learning a local surrogate model to explain the model's prediction under the pertubation the input features, while Kernel SHAP (Lundberg and Lee, 2017) applies the Shapley value concept to quantify the contribution of each input feature to the model's prediction. Influence function methods (Koh and Liang, 2017) trace attribution across the inference-training pipeline and attribute the model's prediction to a specific subset of training data. In concurrent work to ours, Harris et al. (2026) tackle the problem of inference-time content attribution, with the goal of providing formal incentive-compatibility guarantees; they do not require the fairness properties we aim to satisfy. Their approach is based on approximating the least core among candidate sources.

B REWARD ALLOCATION MECHANISMS

We envision reward allocation mechanisms (MAXSHAPLEY or others) could be used in various ways to compensate content providers.

Direct Payment based on Fair Attribution. One straightforward application of MAXSHAPLEY is to use the attributed values as a ratio to allocate a fixed budget to information providers based on their contributions to the final answers. This budget can be funded by either the users (e.g. through a subscription fee) or the generative search providers (e.g. through a fraction of their own ads revenue). A primary challenge would be establishing a payment channel between search providers and content providers; this may be feasible in domain-specific scenarios with limited content providers (e.g., academic publishers, news sites).

Advertisement Proxy based on Fair Attribution. Another possibility is to use a generative search engine to forward advertisements to viewers. That is, the generative search engine can detect the displayed advertisement on the search result pages. Once the attributed values provided by MAXSHAPLEY are obtained, search providers can use the attributed values either as a probability distribution or an auction bid to allocate the advertisement slots to information providers, then show the corresponding advertisements to the users. Hence, content providers can still earn advertisement revenue. This model is (relatively) more backwards compatible with today's web advertisement ecosystem. One potential downside is that advertisements displayed alongside LLM-generated answers may be less effective than in their original form, on their own webpages.

Ad Auction Mechanism based on MAXSHAPLEY Attribution. Finally, MAXSHAPLEY could be combined with other auction-based mechanisms for advertisement allocation. Hajiaghayi et al. (2024) proposed an auction-based mechanism for LLM-generated answers, where each advertiser bids on the opportunity to influence the final response. In their paper, a key technique is to compute the "adjusted bids" for each advertiser based on their bid and also an "attribution score" that is assumed to be available and linearly related to the click-through rate (CTR). The core of their mechanism is a probabilistic second-price auction based on the adjusted bids. (Crucially, this setting requires attribution to be computed by a third party (the advertiser or platform) without access to model internals, making black-box attribution a structural necessity rather than merely a practical convenience.) MAXSHAPLEY could be used to compute the attribution score for each advertiser based on their contribution to the LLM's answer.

C COVERAGE GAMES AND THE MAX-GAME SHAPLEY FORMULA

Here, we derive the max-game Shapley formula used in Section 3.1 through a "coverage-game representation". Intuitively, in a coverage game, each player covers a set of points, and a coalition's utility depends on the number of its players covering each points. The Shapley value divides each point's value equally among the players covering it and each player's total Shapley value will be the sum of these shares over its full coverage area. We establish this rule for general coverage games and then specialize it to a 1-dimensional case to obtain the max-game formula.

C.1 GENERAL COVERAGE GAME

Coverage utility. Let $(\Omega, \mathcal{F}, \mu)$ be a measure space with players indexed by $[m] = \{1, \ldots, m\}$. Each player i covers a measurable set $A_i \subseteq \Omega$. For a coalition $I \subseteq [m]$, define its coverage count at x by

$$h_I(x) = \sum_{i \in I} \mathbf{1}\{x \in A_i\}.$$

A *coverage value function* $g : \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$ specifies the value of covering a point t times. Essentially, $g(t)$ is the "value" of a point when it is covered by $t \in \mathbb{Z}_{\geq 0}$ times. We require $g(0) = 0$, so an uncovered point contributes nothing. For example, $g(t) = \mathbf{1}\{t > 0\}$ means if a point gets covered by at least one time, it's value stays constant (which is used in our later max-coverage game), and $g(t) = t$ corresponds to an additive coverage game.

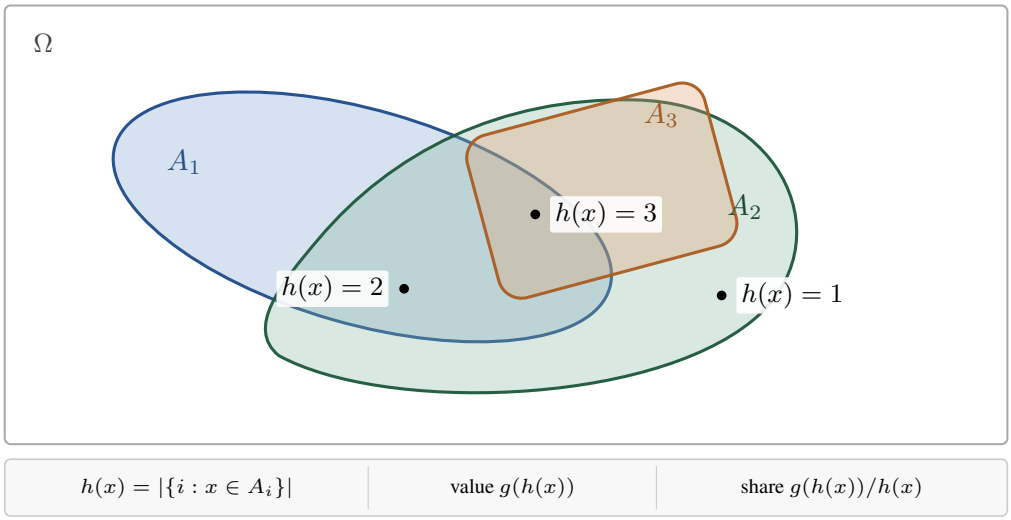

Figure 4: A coverage game with three players and uniform value density. Player i covers A_i. The count $h(x)$ includes all players whose sets contain x. At each covered point, Shapley attribution divides the value $g(h(x))$ equally among these $h(x)$ players.

To allow points to differ in value, let $\rho : \Omega \to \mathbb{R}_{\geq 0}$ be a measurable value density satisfying $\int_{\bigcup_{i=1}^m A_i} \rho(x)\, d\mu(x) < \infty$. The coverage utility for any coalition subset $I \subset [m]$ is

$$U_g(I) = \int_\Omega \rho(x) g(h_I(x))\, d\mu(x). \tag{9}$$

That is, we are integrating all areas covered by I. For each "small surrounding area" around some point x, the coverage count by players in I will be $h_I(x)$, and the corresponding coverage score will be $g(h_I(x))$. Then, the product of its measureable value density $\rho(x)d\mu(x)$ and $g(h_I(x))$ will be the score for this "small surrounding area" around point x. The integrability assumption ensures that every coalition has finite utility. For a finite set of points with counting measure, the integral reduces to a finite sum.

Shapley attribution. At a fixed point x covered by h players, the local utility depends only on which of these players belong to the coalition. The covering players are symmetric, and all other players are null players. The Shapley axioms therefore assign each covering player a share $\rho(x)g(h)/h$. Integrating these local shares gives the following result.

Theorem C.1 (Pointwise Shapley value for coverage games)**.** *Let $h(x) = h_{[m]}(x)$ be the total number of players covering x. For the utility in Equation* (9)*, player i has Shapley value*

$$\phi_i^{U_g} = \int_{A_i} \rho(x) \frac{g(h(x))}{h(x)}\, d\mu(x). \tag{10}$$

The expression is well-defined since $h(x) \geq 1$ on A_i (recall A_i is the area covered by player i).

Proof. By Equation (1), the Shapley value is the expected marginal contribution in a uniformly random ordering of the players. Fix $x \in A_i$. The rank of player i among the $h(x)$ covering players is uniform on $\{1, \ldots, h(x)\}$. At rank r, player i increases the coverage count from $r-1$ to r. Its expected local marginal contribution is therefore

$$\frac{\rho(x)}{h(x)} \sum_{r=1}^{h(x)} \big(g(r) - g(r-1)\big) = \rho(x) \frac{g(h(x))}{h(x)},$$

where the intermediate terms cancel and $g(0) = 0$. At points outside A_i, its contribution is zero. Since there are finitely many player orderings, averaging their marginal contributions and integrating can be done in either order. Integrating the expected local contributions therefore gives Equation (10). ■

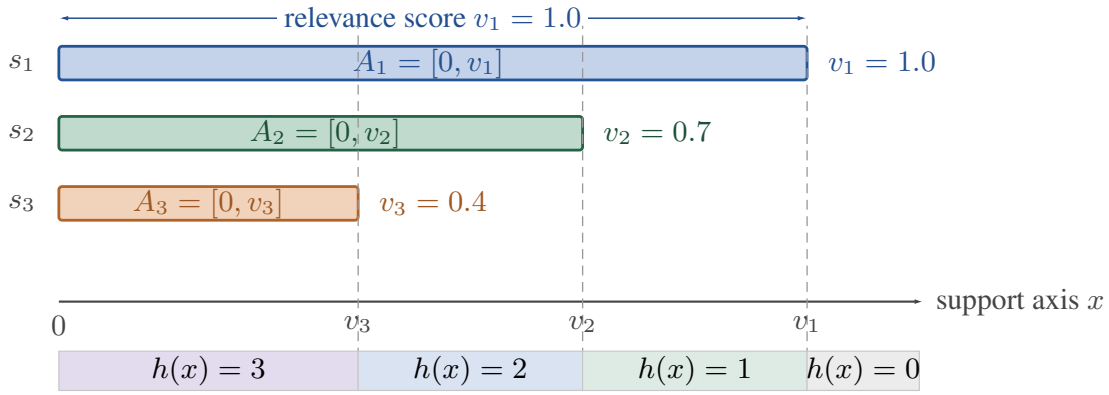

Figure 5: The max game for scores 1.0, 0.7, and 0.4. All three sources share the first segment of length 0.4; two share the next segment of length 0.3; one receives the last segment of length 0.3. Each source's Shapley value is the sum of its shares.

The theorem requires neither monotonicity nor concavity of g. A nondecreasing value function ensures that adding a player cannot reduce utility, while nonincreasing increments give diminishing returns for repeated coverage. These are optional modeling choices.

C.2 The Max Game as a One-Dimensional Coverage Game

Interval representation. Consider one key point with nonnegative source scores $v_1, \dots, v_m$. Represent source i by the interval $A_i = [0, v_i]$. For a nonempty coalition I, the union of its intervals is $[0, \max_{i \in I} v_i]$. Thus its length is exactly the max-game utility.

Formally, take $\Omega = \mathbb{R}_{\geq 0}$ with Lebesgue measure (interval length), $\rho(x) = 1$, and $g(t) = \mathbf{1}\{t > 0\}$. Then

$$U_g(I) = \int_0^\infty \mathbf{1}\{\exists i \in I : x \leq v_i\}\, dx = \max_{i \in I} v_i, \tag{11}$$

where the maximum over the empty set is zero. By Theorem C.1, each source receives an equal share of every interval segment it covers:

$$\phi_i^{\mathsf{Max}} = \int_0^{v_i} \frac{1}{h(x)}\, dx, \qquad h(x) = |\{j : x \leq v_j\}|. \tag{12}$$

Example. Consider the scores $v_1 = 1.0$, $v_2 = 0.7$, and $v_3 = 0.4$ in Figure 5. All three sources receive $0.4/3$ from the first segment. Sources 1 and 2 also receive $(0.7 - 0.4)/2$ each from the next segment, and source 1 alone receives the remaining $1.0 - 0.7$. Their Shapley values are therefore

$$\phi_1^{\mathsf{Max}} = \frac{0.4}{3} + \frac{0.3}{2} + 0.3, \qquad \phi_2^{\mathsf{Max}} = \frac{0.4}{3} + \frac{0.3}{2}, \qquad \phi_3^{\mathsf{Max}} = \frac{0.4}{3}.$$

Exact Shapley for Maximization Game We have the following corollary for the exact Shapley computation for the maximization game.

Corollary C.2 (Exact Shapley computation for the maximization game)**.** *Consider $U_{\mathsf{Max}}(I) = \max_{i \in I} v_i$ with nonnegative scores and $U_{\mathsf{Max}}(\emptyset) = 0$. Relabel the players in sorted order so that $0 \leq v_1 \leq \cdots \leq v_m$, and set $v_0 = 0$. The player at sorted rank r has Shapley value*

$$\phi_r^{\mathsf{Max}} = \sum_{\ell=1}^{r} \frac{v_\ell - v_{\ell-1}}{m - \ell + 1}. \tag{13}$$

All values can be computed in $O(m \log m)$ time.

Proof. Each nonempty segment $(v_{\ell-1}, v_\ell]$ has length $v_\ell - v_{\ell-1}$ and is covered by the $m - \ell + 1$ players at ranks $\ell, \ldots, m$. Each of these players receives a share $(v_\ell - v_{\ell-1})/(m - \ell + 1)$. The player at rank r covers exactly the segments with $\ell \leq r$, giving Equation (13). Tied scores create zero-length segments, which contribute zero.

Sorting costs $O(m \log m)$ time. The values in Equation (13) are prefix sums, so a single scan over the sorted players can computes the corresponding Shapley in $O(m)$ time: each player just adds the marginal term over a running counter. ■

Proof of Proposition 3.1. The utility of MAXSHAPLEY is the weighted sum of n max games, one per key point. The corollary computes the exact Shapley values of each game. Shapley additivity then shows that their weighted sum, as computed by Algorithm 1, is the exact Shapley value of the full utility.

For the complexity bound, Stage I makes nm source-keypoint scoring calls, each with input length at most L. The budget T covers the non-repeated prompts and all generated outputs, including the outputs of these scoring calls. Under the stated per-token cost model, Stage I therefore uses $O((T + nmL)|\Psi|)$ FLOPs. In Stage II, sorting and scanning costs $O(m \log m)$ per key point, and adding the weighted values costs $O(m)$ per key point. The total cost is $O((T+nmL)|\Psi|+nm \log m)$, as reported in Table 1, with no further LLM calls after Stage I.

C.3 REWARDING CORROBORATING INFORMATION

When multiple independent sources support the same key point, their agreement may increase confidence in that key point beyond the support provided by any single source. An attribution rule may therefore assign additional value to this corroboration, allowing the coalition's utility to exceed its highest individual support score. This additional value can then be shared among the supporting sources through Shapley attribution.

The coverage-game model in Equation (9) supports this additional reward through the choice of value function g. For example, we can keep the same source intervals and replace the max-coverage value function with

$$g_\lambda(0) = 0, \qquad g_\lambda(t) = 1 + \lambda(t-1) \quad \text{for } t \geq 1, \qquad \lambda \in [0, 1].$$

At each point, the first covering source contributes one unit of value, and each additional source contributes λ units. Thus $\lambda = 0$ gives the max game, $\lambda = 1$ gives additive coverage, and intermediate values give a smaller reward for additional support. To cap the reward at $c \geq 1$, one may instead use $g(t) = \min\{g_\lambda(t), c\}$.

To model diminishing marginal rewards, we can use exponentially decreasing increments:

$$g_\beta(t) = \frac{1 - \beta^t}{1 - \beta}, \qquad 0 < \beta < 1.$$

This value function satisfies $g_\beta(0) = 0$ and $g_\beta(1) = 1$. At each point, the r-th covering source increases the utility by $g_\beta(r) - g_\beta(r-1) = \beta^{r-1}$. Each additional source therefore contributes a fraction β of the preceding source's increment, and the total local value is bounded by $1/(1-\beta)$. The Shapley formula in Theorem C.1 applies to all these choices without changing the derivation.

Limitations in capturing source dependencies. Agreement among multiple documents need not reflect independent evidence or justify greater confidence in a claim. For example, several articles may trace back to the same original report or repeat a rumor from a common source, even when their wording differs. The coverage count alone does not capture these dependencies.

One possible extension is to use citations, links, and quoted passages to trace the evidence for each key point and group sources that rely on the same original report. Each group could then act as a single player, with its support score defined as the maximum score among its members. Applying the coverage game to these groups would reward support from distinct evidence origins while counting repeated reports within a group only once. This approach would depend on how reliably shared origins can be identified. We leave this extension to future work.

D Economic Implication of Relevance Alignment: Technical Details

D.1 Missing Proofs

Proof of Proposition 4.1. For $B(\boldsymbol{\phi}) > 0$, normalization rescales every payment by the same positive quantity and therefore preserves all pairwise orderings:

$$\phi_i \geq \phi_k \quad \Longleftrightarrow \quad p_i(\boldsymbol{\phi}) \geq p_k(\boldsymbol{\phi}), \quad \forall i, k \in [m]. \tag{14}$$

First, suppose $\boldsymbol{p}(\boldsymbol{\phi}) \in \mathcal{F}(R)$. By the definition of $\mathcal{F}(R)$ and (14), we have $\phi_i \geq \phi_k, \ \forall i \in R, \ k \in [m] \setminus R$, i.e. every relevant source receives at least as much payment as every irrelevant source. Thus, no irrelevant source strictly outranks a relevant source. Since $|R| = K$, the sources in R form an admissible top-K set, where any ties at the cutoff can be resolved in their favor. Hence, there exists a tie-breaking rule under which $\mathrm{Top}_K(\boldsymbol{\phi}) = R$, giving Jaccard@$K$= 1.

Conversely, suppose some top-K tie-breaking rule yields $\mathrm{Top}_K(\boldsymbol{\phi}) = R$. If there existed $i \in R$ and $k \in [m] \setminus R$ with $p_k(\boldsymbol{\phi}) > p_i(\boldsymbol{\phi})$, then $\phi_k > \phi_i$ and k would strictly outrank i, so a top-K set could not include i while excluding k (contradiction). Therefore,

$$p_i(\boldsymbol{\phi}) \geq p_k(\boldsymbol{\phi}), \qquad \forall i \in R, \ k \in [m] \setminus R.$$

Since $\boldsymbol{p}(\boldsymbol{\phi})$ is nonnegative and sums to one by definition, it satisfies all the conditions of $\mathcal{F}(R)$.

Proof of Theorem 4.3. Write $\boldsymbol{p}^{\mathrm{MS}} := \boldsymbol{p}(\boldsymbol{\phi}^{\mathrm{MaxShapley}})$ and $B^{\mathrm{MS}} := B(\boldsymbol{\phi}^{\mathrm{MaxShapley}}) > 0$. If $K = m$, then $R = T = [m]$, so $J = 1$. Moreover, the relevance-order inequalities defining $\mathcal{F}(R)$ are vacuous, so every probability vector belongs to $\mathcal{F}(R)$. Hence, $\mathcal{L}(\boldsymbol{\phi}^{\mathrm{MaxShapley}}) = 0$. Together with $\Delta(V) = 0$, this proves the result for $K = m$. We therefore assume $K < m$ for the remainder of the proof.

For each keypoint j, let $\mathcal{A}_j(t) := \{i : v_{ij} \geq t\}$ for $t \in [0, \max_{i\in[m]} v_{ij}]$. For every nonempty $A \subseteq [m]$, define $\boldsymbol{u}(A)$ by

$$u_i(A) := \frac{\mathbf{1}\{i \in A\}}{|A|}, \qquad i \in [m].$$

Then, Lemma D.1 gives

$$\boldsymbol{p}^{\mathrm{MS}} = \frac{1}{B^{\mathrm{MS}}} \sum_{j=1}^{n} w_j \int_0^{\max_{i\in[m]} v_{ij}} \boldsymbol{u}(\mathcal{A}_j(t))\, dt,$$

$$B^{\mathrm{MS}} = \sum_{j=1}^{n} w_j \max_{i\in[m]} v_{ij}.$$

Note that the coefficients in this representation have total mass

$$\frac{1}{B^{\mathrm{MS}}} \sum_{j=1}^{n} w_j \int_0^{\max_{i\in[m]} v_{ij}} 1\, dt = \frac{1}{B^{\mathrm{MS}}} \sum_{j=1}^{n} w_j \max_{i\in[m]} v_{ij} = 1.$$

Because each $\mathcal{A}_j(t)$ changes only at finitely many support scores, $\boldsymbol{p}^{\mathrm{MS}}$ is therefore a finite convex combination of the vectors $\boldsymbol{u}(\mathcal{A}_j(t))$.

If $t \leq \min_{i\in R} v_{ij}$, every relevant source belongs to $\mathcal{A}_j(t)$. If $t > \max_{k\notin R} v_{kj}$, no irrelevant source belongs to $\mathcal{A}_j(t)$. In either case, $\boldsymbol{u}(\mathcal{A}_j(t)) \in \mathcal{F}(R)$. Conversely, when $\min_{i\in R} v_{ij} < t \leq \max_{k\notin R} v_{kj}$, some relevant source receives zero while some irrelevant source receives a positive share, so $\boldsymbol{u}(\mathcal{A}_j(t)) \notin \mathcal{F}(R)$. Since the loss vanishes on $\mathcal{F}(R)$, while Lemma D.2 bounds the loss of every unit-budget allocation by $K/(K+1)$, we obtain the pointwise bound

$$\mathcal{L}\big(\boldsymbol{u}(\mathcal{A}_j(t))\big) \leq \frac{K}{K+1} \mathbf{1}\left\{ \min_{i\in R} v_{ij} < t \leq \max_{k\notin R} v_{kj} \right\}.$$

Finally, note that $\mathcal{L}(\boldsymbol{\phi}^{\mathrm{MaxShapley}}) = \mathcal{L}(\boldsymbol{p}^{\mathrm{MS}})$ by definition. Applying the convexity from Lemma D.2 to the above convex combination, and then substituting the pointwise bound, gives

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}^{\mathrm{MaxShapley}}) &= \mathcal{L}(\boldsymbol{p}^{\mathrm{MS}}) \\ &\le \frac{1}{B^{\mathrm{MS}}} \sum_{j=1}^{n} w_j \int_0^{\max_{i\in[m]} v_{ij}} \mathcal{L}\big(\boldsymbol{u}(\mathcal{A}_j(t))\big)\, dt \\ &\le \frac{K}{K+1} \frac{1}{B^{\mathrm{MS}}} \sum_{j=1}^{n} w_j \int_0^{\max_{i\in[m]} v_{ij}} \mathbf{1}\left\{ \min_{i\in R} v_{ij} < t \le \max_{k\notin R} v_{kj} \right\} dt \\ &= \frac{K}{K+1} \Delta(V). \end{aligned} \tag{15}$$

We next bound the payment-share gaps. Fix $i \in R$ and $k \notin R$. If $t \le \min_{r\in R} v_{rj}$, then $i \in \mathcal{A}_j(t)$, so $u_i(\mathcal{A}_j(t)) = 1/|\mathcal{A}_j(t)|$, while $u_k(\mathcal{A}_j(t))$ is either $1/|\mathcal{A}_j(t)|$ or zero. If $t > \max_{\ell\notin R} v_{\ell j}$, then $k \notin \mathcal{A}_j(t)$, so $u_k(\mathcal{A}_j(t)) = 0$, while $u_i(\mathcal{A}_j(t)) \ge 0$. Therefore, in either case,

$$u_k(\mathcal{A}_j(t)) - u_i(\mathcal{A}_j(t)) \le 0.$$

For $\min_{r\in R} v_{rj} < t \le \max_{\ell\notin R} v_{\ell j}$, this difference is at most one. Integrating therefore gives

$$\begin{aligned} p_k^{\mathrm{MS}} - p_i^{\mathrm{MS}} &\le \frac{1}{B^{\mathrm{MS}}} \sum_{j=1}^{n} w_j \int_0^{\max_{r\in[m]} v_{rj}} \mathbf{1}\left\{ \min_{r\in R} v_{rj} < t \le \max_{\ell\notin R} v_{\ell j} \right\} dt \\ &= \Delta(V). \end{aligned}$$

Applying Lemma D.3 with $\varepsilon = \Delta(V)$ yields

$$\mathcal{L}(\boldsymbol{\phi}^{\mathrm{MaxShapley}}) \le K\, \Delta(V) \frac{1-J}{1+J}. \tag{16}$$

Combining (15) and (16), we obtain

$$\mathcal{L}(\boldsymbol{\phi}^{\mathrm{MaxShapley}}) \le K\, \Delta(V) \min\left\{ \frac{1-J}{1+J}, \frac{1}{K+1} \right\},$$

which is the required statement.

D.2 Helpful Lemmas

Lemma D.1 (Threshold decomposition of MaxShapley)**.** *For nonnegative support scores and weights, define $\mathcal{A}_j(t) := \{i : v_{ij} \ge t\}$ for $t \in [0, \max_{i\in[m]} v_{ij}]$. Then*

$$\boldsymbol{\phi}^{\mathrm{MaxShapley}} = \sum_{j=1}^{n} w_j \int_0^{\max_{i\in[m]} v_{ij}} \boldsymbol{u}(\mathcal{A}_j(t))\, dt, \qquad B(\boldsymbol{\phi}^{\mathrm{MaxShapley}}) = \sum_{j=1}^{n} w_j \max_{i\in[m]} v_{ij}.$$

Proof. For every coalition $S \subseteq [m]$, MaxShapley's utility satisfies

$$U_{\mathrm{MaxShapley}}(S) = \sum_{j=1}^{n} w_j \max_{i\in S} v_{ij} = \sum_{j=1}^{n} w_j \int_0^{\max_{i\in[m]} v_{ij}} \mathbf{1}\{S \cap \mathcal{A}_j(t) \ne \emptyset\}\, dt.$$

For each keypoint, the indicator is one exactly at the thresholds reached by at least one source in S, giving an integral equal to the maximum support score in that coalition. For $S = \emptyset$, both sides are zero.

At a fixed threshold, consider the game

$$S \longmapsto \mathbf{1}\{S \cap \mathcal{A}_j(t) \ne \emptyset\}.$$

In a uniformly random source ordering, the first source from $\mathcal{A}_j(t)$ makes a marginal contribution of one, and every other source makes a marginal contribution of zero. Each member of $\mathcal{A}_j(t)$ is first with probability $1/|\mathcal{A}_j(t)|$. Therefore, the Shapley allocation of this game is $\boldsymbol{u}(\mathcal{A}_j(t))$.

The active sets change only at finitely many support scores, so the integrals reduce to finite sums. Linearity of the Shapley value consequently gives

$$\boldsymbol{\phi}^{\mathrm{MaxShapley}} = \sum_{j=1}^{n} w_j \int_0^{\max_{i\in[m]} v_{ij}} \boldsymbol{u}(\mathcal{A}_j(t))\, dt.$$

Each integrand is nonnegative and has coordinates summing to one. Hence the allocation is nonnegative, and

$$B(\boldsymbol{\phi}^{\mathrm{MaxShapley}}) = \sum_{j=1}^{n} w_j \int_0^{\max_{i\in[m]} v_{ij}} 1\, dt = \sum_{j=1}^{n} w_j \max_{i\in[m]} v_{ij}.$$

In the above, if $\max_{i\in[m]} v_{ij} = 0$, the corresponding integral is zero. ■

Lemma D.2 (Convexity and a universal loss bound)**.** *Define the probability simplex*

$$\mathcal{P}_m := \left\{ \boldsymbol{p} \in \mathbb{R}^m_{\geq 0} : \sum_{i=1}^{m} p_i = 1 \right\}.$$

The relevance-consistency loss is convex on $\mathcal{P}_m$, and every $\boldsymbol{p} \in \mathcal{P}_m$ satisfies

$$\mathcal{L}(\boldsymbol{p}) \leq \frac{K}{K+1}.$$

Proof. For $\boldsymbol{p} \in \mathcal{P}_m$, normalization leaves $\boldsymbol{p}$ unchanged, so

$$\mathcal{L}(\boldsymbol{p}) = \frac{1}{2} \min_{\boldsymbol{p}^0 \in \mathcal{F}(R)} \|\boldsymbol{p} - \boldsymbol{p}^0\|_1.$$

The set $\mathcal{F}(R)$ contains the uniform vector and is a closed, convex subset of $\mathcal{P}_m$. It is therefore nonempty and compact, so the minimum is attained.

Let $\boldsymbol{p}, \boldsymbol{q} \in \mathcal{P}_m$, and choose minimizers $\boldsymbol{p}^0, \boldsymbol{q}^0 \in \mathcal{F}(R)$ for their respective losses. For any $\lambda \in [0, 1]$, convexity of $\mathcal{F}(R)$ gives

$$\lambda \boldsymbol{p}^0 + (1-\lambda)\boldsymbol{q}^0 \in \mathcal{F}(R).$$

Consequently,

$$\begin{aligned} \mathcal{L}\big(\lambda \boldsymbol{p} + (1-\lambda)\boldsymbol{q}\big) &\leq \frac{1}{2} \left\| \lambda(\boldsymbol{p} - \boldsymbol{p}^0) + (1-\lambda)(\boldsymbol{q} - \boldsymbol{q}^0) \right\|_1 \\ &\leq \lambda \mathcal{L}(\boldsymbol{p}) + (1-\lambda)\mathcal{L}(\boldsymbol{q}), \end{aligned}$$

which proves convexity.

For the universal bound, let $\boldsymbol{e}_i$ allocate the entire unit budget to source i. If $i \in R$, then $\boldsymbol{e}_i \in \mathcal{F}(R)$ and $\mathcal{L}(\boldsymbol{e}_i) = 0$. If $i \notin R$, the allocation $\boldsymbol{u}(R \cup \{i\})$ belongs to $\mathcal{F}(R)$, so

$$\begin{aligned} \mathcal{L}(\boldsymbol{e}_i) &\leq \frac{1}{2} \|\boldsymbol{e}_i - \boldsymbol{u}(R \cup \{i\})\|_1 \\ &= \frac{1}{2} \left(\left| 1 - \frac{1}{K+1} \right| + \sum_{r \in R} \left| 0 - \frac{1}{K+1} \right| \right) \\ &= \frac{1}{2} \left(\frac{K}{K+1} + \frac{K}{K+1} \right) \\ &= \frac{K}{K+1}. \end{aligned}$$

Since every $\boldsymbol{p} \in \mathcal{P}_m$ satisfies $\boldsymbol{p} = \sum_i p_i \boldsymbol{e}_i$, convexity yields

$$\mathcal{L}(\boldsymbol{p}) \leq \sum_{i=1}^{m} p_i \mathcal{L}(\boldsymbol{e}_i) \leq \frac{K}{K+1} \sum_{i \notin R} p_i \leq \frac{K}{K+1}.$$

■

Lemma D.3 (Payment-gap bound)**.** *Let ϕ be a nonnegative payment allocation with $B(\phi) > 0$. Suppose that, for some $\varepsilon \geq 0$,*

$$p_k(\phi) - p_i(\phi) \leq \varepsilon \qquad \forall i \in R,\ k \notin R.$$

For any top-K set T selected under an arbitrary tie-breaking rule, let $J = |T \cap R|/|T \cup R|$. Then

$$\mathcal{L}(\phi) \leq K\varepsilon \frac{1-J}{1+J}.$$

Proof. Write $\boldsymbol{p} = \boldsymbol{p}(\phi)$. Since normalization by a positive budget preserves ordering, T is also a top-K set for $\boldsymbol{p}$. Define

$$M := R \setminus T, \qquad Q := T \setminus R, \qquad b := |M| = |Q|.$$

The cardinalities agree because $|T| = |R| = K$.

Choose any bijection $\pi : M \to Q$ and exchange the payment shares within each pair:

$$p_i^0 = p_{\pi(i)}, \qquad p_{\pi(i)}^0 = p_i \quad (i \in M), \qquad p_\ell^0 = p_\ell \quad (\ell \notin M \cup Q).$$

This permutation preserves nonnegativity and the unit budget. After the exchanges, the shares assigned to R are precisely the original shares $\{p_t : t \in T\}$: sources in $R \cap T$ retain their shares, while each $i \in M$ receives the share of $\pi(i) \in Q$. Similarly, the shares assigned to $[m] \setminus R$ are precisely $\{p_\ell : \ell \notin T\}$. Since T is a top-K set, $p_t \geq p_\ell$ for every $t \in T$ and $\ell \notin T$. Therefore, $p_i^0 \geq p_k^0$ for every $i \in R$ and $k \notin R$, so $\boldsymbol{p}^0 \in \mathcal{F}(R)$.

In the top-K selection for T, each $\pi(i)$ is selected while $i \in M$ is not, giving $p_{\pi(i)} \geq p_i$. Since the exchanged pairs are disjoint,

$$\begin{aligned}
\mathcal{L}(\phi) &\leq \frac{1}{2}\|\boldsymbol{p} - \boldsymbol{p}^0\|_1 \\
&= \frac{1}{2}\sum_{i \in M} \left(|p_i - p_{\pi(i)}| + |p_{\pi(i)} - p_i|\right) \\
&= \sum_{i \in M} (p_{\pi(i)} - p_i) \\
&\leq \sum_{i \in M} \varepsilon = b\varepsilon. \qquad (17)
\end{aligned}$$

When $b = 0$, no exchanges are needed, and the same inequality holds with an empty sum.

Finally, $|T \cap R| = K - b$ and $|T \cup R| = K + b$, so

$$J = \frac{K-b}{K+b}, \qquad b = K\frac{1-J}{1+J}. \qquad (18)$$

Substituting (18) into (17) proves the claim. ∎

E Experimental Setup

E.1 Full Evaluation Pipeline

Each dataset example consists of a query q and an ordered set of retrieved sources $S = (s_1, \ldots, s_m)$, where each source is a text snippet. Some datasets also provide a reference answer $\tilde{a}$ for evaluation; in our experiments, this reference answer is available for HotpotQA and MuSiQUE.

For each example, we first run the search LLM Ψ once on the full context:

$$a = \Psi(q, S).$$

The generated answer a is the object being attributed. That is, all source credit scores are intended to explain the displayed answer a, not the reference answer $\tilde{a}$.

We then run each attribution method on the same downstream attribution instance: the query q, the sources $S = (s_1, \ldots, s_m)$, and the generated answer a. Different methods obtain scores in different

ways. Some baselines repeatedly evaluate ablated source subsets, while MAXSHAPLEY uses an attribution LLM Ψ_A as a scoring tool inside its attribution algorithm. To isolate differences between attribution methods rather than differences between language models, we use the same underlying LLM for Ψ_A as for the search LLM Ψ in each model setting.

We treat FullShapley as the attribution-quality reference. When a dataset provides a reference answer $\tilde{a}$, we additionally include $\tilde{a}$ in the LLM-judge prompt used to evaluate utilities inside the FullShapley pipeline. This oracle information is used only to obtain a high-fidelity reference attribution; it does not change the generated answer a being attributed.

After each method produces source scores, we evaluate them in two ways. First, we compare the selected or top-ranked sources against human source annotations using Jaccard index. Second, we compare each method's full ranking of source credits against FullShapley using Kendall's τ_b.

We repeat this end-to-end evaluation pipeline three times and report aggregate metrics across runs, reducing variance from stochastic generation and LLM-based scoring.

E.2 LLM PROMPTS

In LLM-as-a-judge, we use the Attribution LLM (Figure 1) to compute the Judge function, using both GPT-4.1 nano and Haiku 4.5. The LLM Ψ_A receives a query, an LLM-generated response, a subset of the information sources, and, if the ground truth is available, the ground truth to the answer, then produces a score indicating how well that subset answers the query. We designed our prompts to elicit reliable quality assessments while preventing knowledge hallucination—ensuring the model relies solely on provided sources rather than its parametric knowledge. Scores range from 0.0 to 1.0, where 1.0 indicates a perfect answer addressing all parts of the question, and 0.0 indicates an incorrect or unsupported answer. The full prompt is provided in Figure 6; we provide ground truth only to FullShapley and approximation baselines.

For MAXSHAPLEY, in addition to the basic LLM-generated response prompt for answering the query with a set of information sources, we used an LLM to break the response down into key points, then to "distill" these key points (filtering out repetitive or redundant key points, see Appendix G for details), and then asked the LLM to rank each information source in terms of relevance to each key point, which serves as our value function for MAXSHAPLEY. We include the full version of the keypoint breakdown prompt in Figure 7, the distillation prompt in Figure 9, and the relevancy scoring in Figure 8.

```
You are an evaluation system that evaluates AI responses to a question. The goal is to
    determine if the response is correct and well-founded.


EVALUATION FOCUS:
Judge the AI's final answer and how it got to its answer. Correct answers built upon evidence
    are better than correct guesses. Reasonable inference and pulling together well-founded
    pieces of information is good. It's crucial to consider what the question is asking as
    well.


Question: {input query}
Ground Truth: {input ground_truth, when available}
AI Response: {input LLM-generated response}


Does the AI response fully address the question and convey the same meaning as \"{input LLM-
    generated response}\" with confirmed information?


Justification: [Does the AI response fully address the question and convey the same meaning as
     the ground truth with confirmed information? ]
Score: [Number from 0.0 to 1.0]
```

Figure 6: Simplified LLM-as-a-judge prompt, FullShapley.

E.3 KEYPOINT EXAMPLES

We provide an example of two keypoints resulting from the answering of one query below.

```
You are an evaluation system that evaluates AI responses to a question. The goal is to
     determine if the response is correct and well-founded.


EVALUATION FOCUS:
Judge the AI's final answer and how it got to its answer. Correct answers built upon evidence
     are better than correct guesses. Reasonable inference and pulling together well-founded
     pieces of information is good. It's crucial to consider what the question is asking as
     well.


Question: {input query}
Ground Truth: {input ground_truth, when available}
AI Response: {input LLM-generated response}


Does the AI response fully address the question and convey the same meaning as \"{input LLM-
     generated response}\" with confirmed information?


Justification: [Does the AI response fully address the question and convey the same meaning as
      the ground truth with confirmed information? ]
Score: [Number from 0.0 to 1.0]
```

Figure 7: Sketch, MAXSHAPLEY keypoint breakdown prompt.

```
You are evaluating if a source document provides substantive support for a specific statement.

SCORING SCALE (0.0 to 1.0):
- 0.0 = No Support: Source lacks information to support the statement, even if same topic
- 0.3 = Minimal Support: Source has some relevant info but missing key details
- 0.7 = Substantial Support: Source contains most information needed, minor gaps only
- 1.0 = Complete Support: Source explicitly contains all information to support the statement

RESPONSE FORMAT:
EXPLANATION: [2 sentences on what specific information the source does/doesn't provide]
SCORE: [0.0 to 1.0]

Evaluate how well this source supports the statement:
SOURCE: {source}
STATEMENT: {keypoint}
```

Figure 8: Keypoint relevance scoring prompt, MAXSHAPLEY.

E.4 BASELINE PSEUDOCODE

Here, we provide the pseudocode for the baselines that we used in the experiments. In Algorithm 2, we include the brute-force algorithm for computing Shapley value. While there exist more efficient approximations to the Shapley value, the exact computation is known to have exponential complexity.

```
You are evaluating whether a source document provides substantive informational support for a
     specific statement.
CRITICAL: Being on the same topic is not sufficient. The source must contain specific
     information that directly supports the statement's claims.
Semantic equivalence or clear logical entailment is allowed. Reasonable and clear
     interpretation is also allowed - for example, if the statement refers to rectangles and
     the source refers to squares, that counts as support since the claim logically applies.

SCORING SCALE (0.0 to 1.0):
0.0 = No Support: Source lacks information to support the statement, even if on the same topic
     .
0.3 = Minimal Support: Source has some relevant information but is missing key details.
0.7 = Substantial Support: Source contains most of the information needed, with only minor
     gaps.
1.0 = Complete Support: Source explicitly contains all information required to support the
     statement.

KEY RULE:
Only score based on substantive informational support, not topical similarity.
Statements about what is not mentioned should score 0.0.
```

Figure 9: Keypoint distillation prompt, MAXSHAPLEY.

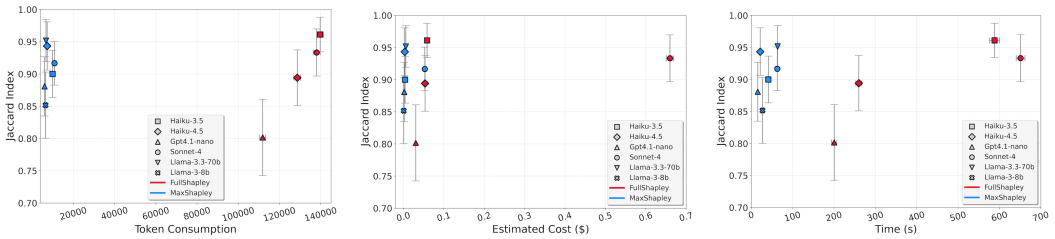

Figure 10: Attribution quality versus (left) token consumption, (center) cost of computation, and (right) computation time across six models and two Shapley algorithms. Haiku 3.5 and 4.5 modestly outperform GPT-4.1 nano and Llama 3 in quality at slightly higher token costs, computation costs, and computation time. Sonnet 4's increased capabilities, costs, and computation time do not translate to quality improvements. Llama 3.3's increased capabilies do result in quality improvements, at cost comparable with Sonnet 4.

```
Query: What government position was held by the woman who portrayed Corliss Archer
in Kiss and Tell?

> Keypoint 1: Shirley Temple portrayed Corliss Archer in the 1945 film.
> Keypoint 2: Shirley Temple served as Chief of Protocol of the United States
```

Figure 11: Keypoint example.

Next, we provide pseudocode for the Monte Carlo Approximation of Shapley Value via Sampling, in Algorithm 3.

E.5 Dataset Annotation

We independently annotated subsets of 100 queries for HotPotQA and MuSiQUE, and 95 queries for MS-MARCO, the maximum number of samples with existing relevance annotations. Each query contains six candidate information sources, selected to include both relevant and irrelevant sources according to the original, noisy dataset labels. Two annotators labeled per-source relevance and then discussed disagreements to reach consensus. Inter-rater reliability before discussion was 85% for HotPotQA, 86% for MS-MARCO, and 92% for MuSiQUE.

Figure 12 shows the cumulative distribution functions of Jaccard index scores measuring agreement between our consensus annotations and the original dataset annotations for HotPotQA, MuSiQUE, and MS-MARCO. The Jaccard index quantifies the overlap between the sets of sources labeled as relevant. HotPotQA and MuSiQUE have binary annotations. For MS-MARCO, which uses a 0-3 relevance scale, we considered sources with scores of 2 or 3 as relevant. We had high agreement with annotations for HotPotQA and MuSiQUE, while MS-MARCO has moderate agreement.

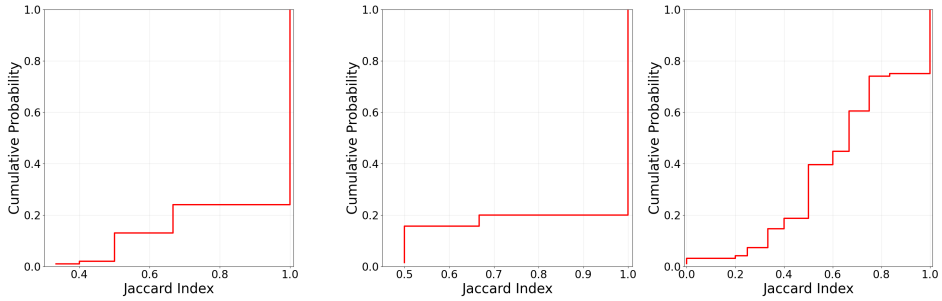

Figure 12: Cumulative distribution functions of Jaccard index scores measuring the overlap between relevant information source sets identified by our consensus annotations and those specified in the original dataset annotations. Jaccard indices were computed on the annotated subset for each dataset (HotPotQA left, MuSiQUE middle, MS-MARCO right). HotPotQA and MuSiQUE exhibit high agreement, with more than half the samples achieving perfect agreement. MS-MARCO shows 25% are in perfect agreement.

Algorithm 2 Full Shapley

Require: A value function $V(\cdot)$ and a set of m elements (e.g., information sources) $S = \{s_1, s_2, \ldots, s_m\}$.
Ensure: Shapley values ϕ_i for each $i \in \{1, \ldots, m\}$.
  Initialize $\phi_i \leftarrow 0$ for all $i \in \{1, \ldots, m\}$.
  for $i \in \{1, \ldots, m\}$ **do**
    for $j \in \{0, \ldots, m-1\}$ **do**
      Let $\mathcal{T}_j$ be all subsets of size j from $\{1, \ldots, m\} \setminus \{i\}$.
      for each $T \in \mathcal{T}_j$ **do**
        $T' \leftarrow T \cup \{i\}$ {Add element i into subset T}
        $v_{\text{with}} \leftarrow V(T')$
        $v_{\text{without}} \leftarrow V(T)$
        $\Delta \leftarrow v_{\text{with}} - v_{\text{without}}$ {Marginal contribution of source i}
        $\phi_i \leftarrow \phi_i + \dfrac{\Delta}{\binom{m-1}{j} \cdot m}$
      end for
    end for
  end for
  return $\{\phi_i\}_{i \in [m]}$

Algorithm 3 Monte-Carlo Approximation of Shapley Values via Sampling

Require: A value function $V(\cdot)$, number of information sources m, and sample size n.
Ensure: Approximated Shapley values ϕ_i for each $i \in \{1, \ldots, m\}$.
  Initialize $\phi_i \leftarrow 0$ for all $i \in \{1, \ldots, m\}$.
  Let $v_\emptyset \leftarrow V(\emptyset)$ {Value of the empty subset}
  for $r = 1$ to n **do**
    Sample a random permutation π of $\{1, \ldots, m\}$ from the uniform distribution.
    Initialize $T \leftarrow \emptyset$, $v_{\text{prev}} \leftarrow v_\emptyset$
    for i in π **do**
      Let $T' \leftarrow T \cup \{i\}$
      $v_{\text{new}} \leftarrow V(T')$
      $\Delta \leftarrow v_{\text{new}} - v_{\text{prev}}$ {Marginal contributions}
      Update $T \leftarrow T'$, $v_{\text{prev}} \leftarrow v_{\text{new}}$
    end for
  end for
  return $\{\phi_i\}_{i \in [m]}$

F Empirical Evaluation of the Relevance-Consistency Loss

We provide an empirical evaluation of the relevance-consistency loss in (7) and the MaxShapley upper bound in (8) (Theorem 4.3). We evaluate 30 queries from each of HotPotQA, MuSiQUE, and MS-MARCO, with six retrieved sources and three runs per query. Consequently, each MaxShapley variant produces 90 allocations per dataset and 270 allocations across the three datasets.

Setup. We evaluate MaxShapley with GPT-4.1-nano and two additional variants using GPT-5.6-Sol at low reasoning effort. In the *support-scoring* variant, only the support-scoring stage is rerun with GPT-5.6-Sol. The *full-pipeline* variant instead uses GPT-5.6-Sol for answer generation, keypoint extraction and refinement, and support scoring, while retaining the same queries, sources, prompts, and uniform keypoint-weighting rule. We also evaluate FullShapley, LOO, MCU, MCA, and KernelSHAP allocations.

For each nonnegative allocation with positive total budget, we normalize the payments and compute the relevance-consistency loss $\mathcal{L}(\boldsymbol{\phi})$ in (7). We exclude zero-budget allocations, for which the payment shares are undefined, and cases with $K = 0$, for which Jaccard@K and the corresponding bound are undefined. When payments are tied at the top-K cutoff, we use the admissible top-K set with the largest Jaccard@K, yielding the tightest upper bound in (8).

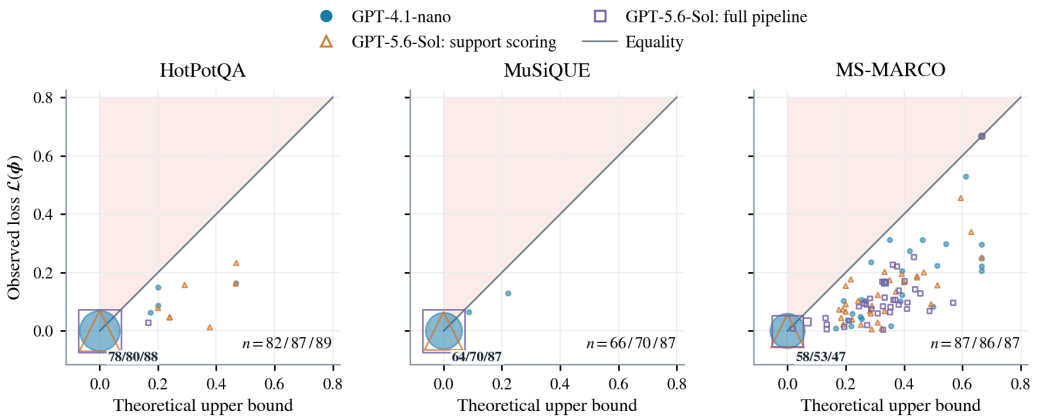

Figure 13: Observed relevance-consistency loss versus the upper bound in (8) for the three MaxShapley variants. Marker area is proportional to the number of allocations with identical coordinates. The origin counts and sample sizes n are reported in the following order: GPT-4.1-nano, GPT-5.6-Sol support scoring, and GPT-5.6-Sol full pipeline. The shaded region above the equality line represents violations of the bound; no allocation lies in this region.

Validation of the upper bound. Figure 13 compares the observed loss with the bound for each valid MaxShapley allocation. Of the 270 saved allocations for each model, 235 GPT-4.1-nano, 243 GPT-5.6-Sol support-scoring, and 263 GPT-5.6-Sol full-pipeline allocations are valid. The bound holds in all 741 comparisons. The loss is zero for 625 of the 741 allocations. Of the remaining 116, 102 are from MS-MARCO, 12 from HotPotQA, and 2 from MuSiQUE; all lie strictly below the upper bound.

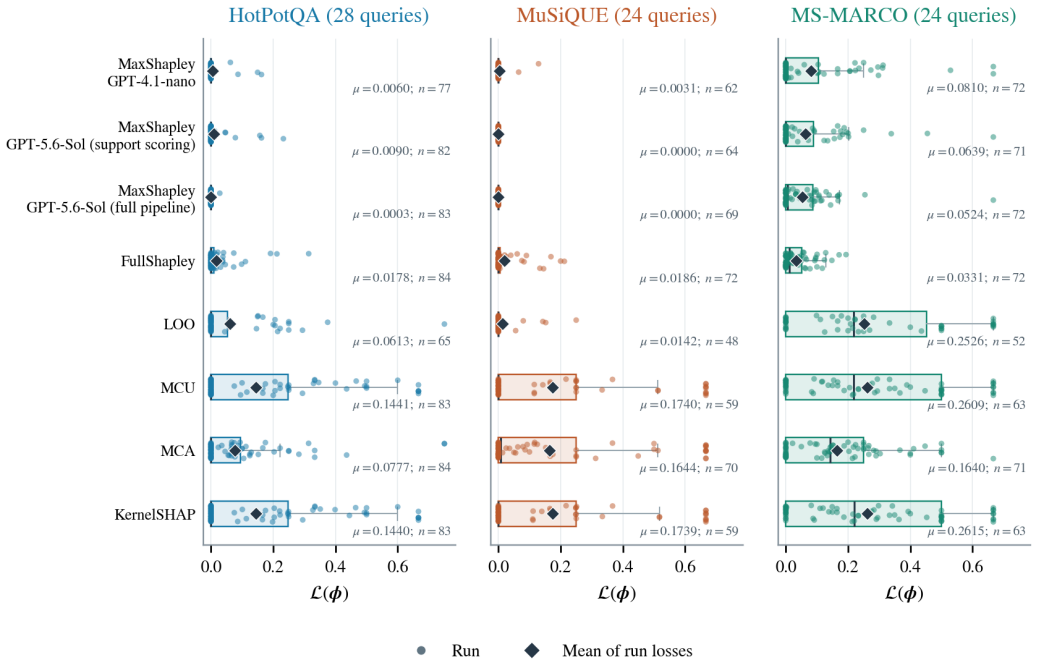

Figure 14: Relevance-consistency loss across attribution methods. Each point is one valid run; diamonds denote means, and n gives the number of valid runs. Boxes show the median and interquartile range, with whiskers extending to 1.5 times the interquartile range.

Loss across attribution methods. Figure 14 compares all eight methods on queries for which every method produces at least one positive-payment allocation, resulting in 28 HotPotQA, 24 MuSiQUE, and 24 MS-MARCO queries. All three MaxShapley variants have near-zero mean loss on HotPotQA and MuSiQUE. On MS-MARCO, replacing GPT-4.1-nano with GPT-5.6-Sol reduces the mean loss

from 0.0810 to 0.0639 when only support scoring is rerun and to 0.0524 when the full pipeline is rerun; FullShapley attains a mean loss of 0.0331.

G ABLATIONS

Response	Mean Quality Score	Std. Dev
Based on the provided sources, I can confidently state: Connie May Fowler was definitely a memoirist. The source 'Connie May Fowler' explicitly states that she wrote memoirs, specifically mentioning "When Katie Wakes" (which explores her family's generational cycle of domestic violence) and "A Million Fragile Bones" (about her life on a barrier island and the Deepwater Horizon oil spill).	0.3	0.0
Based on the provided sources, I can confidently state: Connie May Fowler was a memoirist. The source 'Connie May Fowler' explicitly describes her as a "memoirist" and mentions two of her memoirs: "When Katie Wakes" (which explores her family's generational cycle of domestic violence) and "A Million Fragile Bones" (about her life on a barrier island and the Deepwater Horizon BP oil spill).	1.0	0.0

Table 2: The LLM-as-a-judge Quality evaluation introduces sensitivity to token-level variations in semantically equivalent responses. Response 1 (top) was generated from four relevant sources. Response 2 (bottom) included one additional irrelevant source. Despite being semantically equivalent, the LLM-as-a-judge (Attribution LLM) assigned Quality scores of 1.0 and 0.3 (scale: 0.0-1.0). The consistent scoring across 10 runs suggests that the LLMs are *sensitive* to wording, but *consistent* for the same wording.

Model Selection.

We evaluated three large language models for suitability, GPT-4.1 nano (OpenAI, 2025), Claude Haiku 3.5, and Claude Sonnet 4 (Anthropic, 2025; 2024), but conducted our main experiments using only the first two. As expected, attribution quality improved with model capability: Claude Haiku 3.5 achieved notably higher quality scores than GPT-4.1 nano at comparable token consumption levels across all Shapley algorithms (Figure 10). However, the progression from Haiku 3.5 to Sonnet 4 deviated from this trend. While Sonnet 4 demonstrated greater token efficiency, it did not yield the anticipated improvement in attribution quality.

Investigation revealed that our prompts, optimized for GPT-4.1 nano and Haiku 3.5, proved overly restrictive for Sonnet 4. Specifically, instructions designed to prevent knowledge hallucination (e.g., directing the model not to fill knowledge gaps when sources cannot answer the query) were interpreted too strictly by Sonnet 4, causing it to refuse answering even when sources contained sufficient information. This suggests that prompt engineering requires model-specific calibration. More critically, Sonnet 4's higher cost–an order of magnitude greater than both GPT-4.1 nano and Haiku 3.5 (Figure 10)–combined with the extensive prompt re-engineering required, led us to exclude it from our main experiments.

To further assess model generalizability, we also evaluated two open-source Llama models on MAXSHAPLEY: a larger Llama 3.3-70B-Instruct-Turbo (Grattafiori et al., 2024) and a smaller Llama-3-8B-Instruct-Lite (Grattafiori et al., 2024). The larger model achieved quality improvements comparable to those seen with stronger closed-source models, but at a cost on par with Sonnet 4, making it similarly impractical for our purposes. The smaller model underperformed even GPT-4.1 nano, suggesting that below a certain capability threshold, open-source models are not yet competitive for this task.

When we expanded our annotated dataset from 30 to 100 samples, we re-evaluated using GPT-4.1 nano and Claude Haiku 4.5 (Anthropic, 2024), as Haiku 3.5 had been retired in the interim. Haiku 4.5 outperforms its predecessor across all measured dimensions: despite carrying higher per-token pricing, it consumes fewer tokens overall, resulting in lower total cost, and is generally faster. In terms of attribution quality, Haiku 4.5 also improves over Haiku 3.5, placing it between GPT-4.1 nano and Haiku 3.5 in the overall model landscape (Figure 10). Cost differences between GPT-4.1 nano and Haiku 4.5 remained modest, while GPT-4.1 nano retained its order-of-magnitude speed advantage per sample (Figure 10). While API latency affects these measurements, the consistency of this difference suggests genuine efficiency advantages for time-sensitive applications.

Clipping. When comparing all attribution scores to ground truth relevance labels via Jaccard index, clipping has a minimal effect, with the largest difference being a 0.05 increase for FullShapley on HotPotQA with GPT-4.1 nano. However, clipping substantially improves Kendall τ_b ordinal correlation scores. Extremely small non-zero attribution scores (e.g., < 0.001) introduce noise into ordinal correlation calculations by being treated as distinct ranked values rather than ties. Clipping eliminates this noise by setting near-zero attributions to exactly zero, resulting in clearer ordinal relationships. The most significant improvement was with MuSiQUE with Haiku 3.5, where the ordinal correlation between MAXSHAPLEY and FullShapley increased by 0.113 with clipping applied.

Caching. We used caching in our baseline implementations to improve efficiency. For both FullShapley and the approximation baselines, we cached tested coalitions of sources and reused their LLM-as-a-judge scores upon cache hits to reduce costly LLM API calls. In FullShapley, caching was applied to sorted coalitions of sources–unlike the unsorted caching used in the approximation algorithms–which reduced redundant evaluations and required fewer coalition tests overall. This design choice improved cost and time efficiency: in an experiment with MuSiQUE, Haiku 3.5, and 10 data samples, unsorted caching resulted in a $3\times$ increase in token consumption, runtime, and therefore cost.

Experiments on Large Datasets. We conducted MAXSHAPLEY on the full MuSiQUE and HotPotQA dev datasets, and the MS-MARCO passages dataset with TREC 2019/2020 annotated datasets with GPT-4.1 nano, restricting our analysis to answerable queries (i.e., queries for which the provided information sources contain sufficient information to generate an answer). Figure 15 shows the cumulative distribution of Jaccard index scores across all 2,417 data samples for MuSiQUE, 7,405 data samples for HotPotQA, and a combined 96 data samples for MS-MARCO. We observe a similar pattern to the agreement with our manually-annotated dataset, with more noise in the HotPotQA and MS-MARCO full datasets (this is expected, as we noted the original datasets often had noisy annotations, hence why we manually re-annotated a subset). We observe a slightly noisier Jaccard index on the full MuSiQUE dataset, relative to our manually annotated subset. Although our manual annotations aligned completely with the original dataset labels, our annotated subset consisted primarily of 2-hop reasoning questions. When we evaluated the full MuSiQUE dataset, it also included 3-hop, 4-hop, and 5-hop questions, for which we observed a degradation in the average Jaccard index. This trend is consistent with prior observations that LLMs may exhibit reduced performance as the required reasoning depth increases (Lee et al., 2025), although our experiment does not isolate the specific source of this degradation. Nonetheless, the average Jaccard index for the full MuSiQUE development set remains ≥ 0.70.

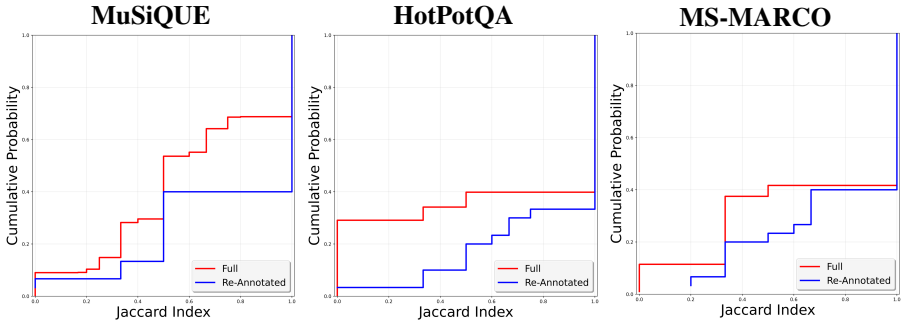

Figure 15: Cumulative distribution function of Jaccard index scores between relevant information sources identified by MAXSHAPLEY and ground truth annotations from the full MuSiQUE answerable dataset (2,417 samples), HotPotQA dev dataset (7,405 samples), and MS-MARCO passages dataset with TREC 2019/2020 relevancy annotations (96 samples) with GPT-4.1 nano. The annotated data set results (on 30 data samples) are also depicted for comparison.

Impact of Keypoint Decomposition. In our current implementation of keypoint decomposition, our prompt has a "keypoint distillation" component, which filters out repetitive or redundant keypoints. To test the robustness of MAXSHAPLEY with different keypoint decomposition methodologies, we test MAXSHAPLEY on our 30-sample manually-annotated datasets with GPT-4.1 nano using the prompt from Figure 7 without the distillation component from Figure 9. The average Jaccard index changes by 0.02-0.13 across datasets. On MuSiQUE (Figure 16), our results improve due to no

distillation (0.13 increase). However, MS-MARCO and HotPotQA, which are more representative of "messy" real-world web queries, suffer slightly (0.02-0.05 reduction) without distillation. This suggests that distillation is (slightly) helping the performance of MAXSHAPLEY. The robustness of MAXSHAPLEY in the face of different keypoint decomposition methodologies—including against adversarial manipulation—remains a direction for future research.

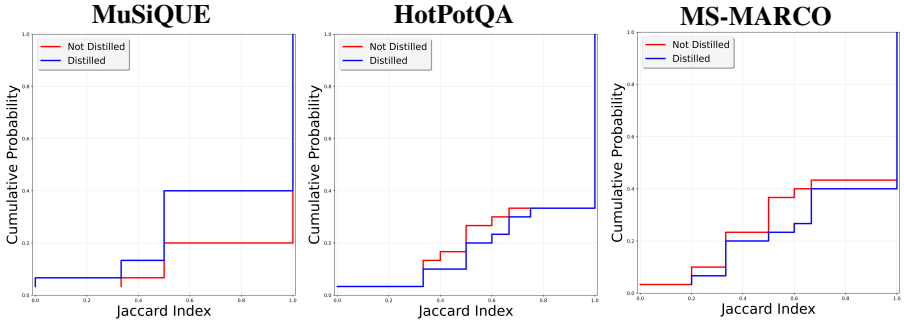

Figure 16: Cumulative distribution function of Jaccard index scores between relevant information sources identified by MAXSHAPLEY and ground truth annotations between keypoint composition that are "distilled" and "not distilled" with GPT-4.1 nano. The average Jaccard index of MuSiQUE increases from 0.76 (distilled) to 0.89 (not distilled), HotPotQA declines from 0.83 to 0.81, and MS-MARCO declines from 0.78 to 0.73.

Scaling with Information Sources A practical concern for MAXSHAPLEY is how computational cost scales with the number of information sources. We evaluate the scalability of MAXSHAPLEY against KernelSHAP (using 5 permutations of data) on MuSiQUE with GPT-4.1 nano, varying the number of sources (see Figure 17).

MAXSHAPLEY exhibits linear scaling in token consumption with respect to the number of sources, whereas KernelSHAP grows at a super-linear rate. While absolute costs may still be prohibitive for some production systems, understanding the scaling behavior of attribution methods is an important step toward assessing their feasibility in real-world deployments.

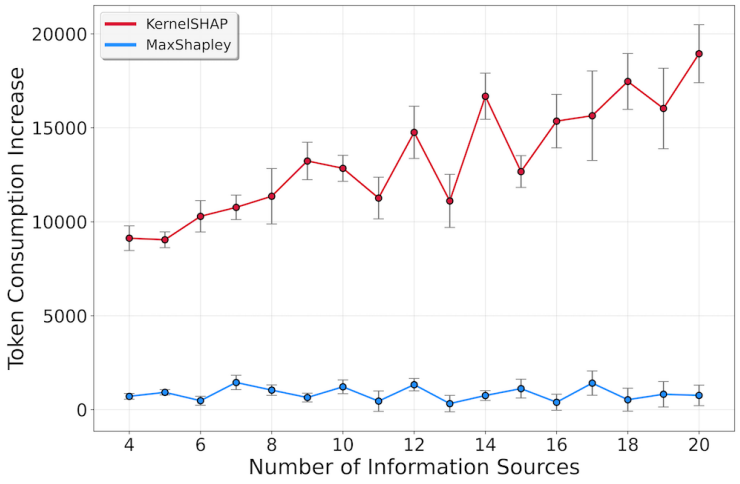

Figure 17: Token consumption increase per additional information source for MAXSHAPLEY and KernelSHAP (calculated with 5 permutations of data), evaluated on MuSiQUE with GPT-4.1 nano. MAXSHAPLEY's marginal token consumption remains approximately constant as sources are added, indicating linear overall scaling. KernelSHAP's marginal cost grows with the number of sources, indicating super-linear scaling.

Additional Baselines: Sentence-Level Leave-One-Out In our evaluation of MAXSHAPLEY, we use source-level Leave-One-Out (LOO) as a baseline, computing each source's contribution as $\phi_i^{\text{LOO}} = U(S) - U(S \setminus \{s_i\})$. A natural variant is to compute attribution at the sentence level, removing individual sentences rather than entire sources: $\phi_i^{\text{LOO}} = \frac{1}{|s_i|} \sum_{t_j \in s_i} [U(S) - U(S \setminus \{t_j\})]$.

We evaluated this variant on HotPotQA with GPT-4.1 nano. Sentence-level LOO underperforms source-level LOO in attribution quality (Jaccard index of 0.26 vs. 0.52) while consuming significantly more tokens (50k vs. 15k). We therefore exclude it as a baseline in our main experiments.

H Implementation Considerations

The actual implementation of Algorithm 1 can vary depending on the use scenario:

- **One-pass or Multiple Pass.** We can either ask the LLM to generate key points and scores in one pass within the same call during the answer generation process, or use multiple calls to the LLM to generate key points and scores separately. As in prior work, we used multiple calls to reduce hallucinations (Gu et al., 2025).
- **Model Selection.** Given that the capability required for the LLM to generate key points and scores is weaker than the complete answer generation process, we can choose a fine-tuned LLM model or a smaller model for different purposes to further reduce computation cost. Our algorithm is designed to be agnostic to model selection; we show ablations in Appendix G.
- **Prompt Customization.** The prompts used in different stages can be customized to further improve performance under different use scenarios. In the generative search scenario, we can even adaptively generate score standards based on the user's query and retrieved sources to further improve score fidelity. We include the prompts for our implementation in Appendix E.2.

I Reproducibility

To facilitate reproducibility and future research, we provide:

- **Open-source code**: Full implementation of MaxShapley and all baselines, including evaluation scripts, all LLM prompts, and experiment configurations.[4]
- **Educational demo**: An interactive web application demonstrating MaxShapley on sample queries.[5]
- **Annotated datasets**: Manually re-annotated subsets of HotPotQA, MuSiQUE, and MS-MARCO (100 queries each) with consensus labels from independent annotators.

[4] https://github.com/anonymousresearcher298/FairAttribution

[5] https://fair-search.com