Diagnosing and Repairing Citation Failures in Generative Engine Optimization

Zhihua Tian[1*†] **Yuhan Chen**[1*] **Yao Tang**[2*] **Jian Liu**[2] **Ruoxi Jia**[1]

[1]Virginia Tech [2]Zhejiang University

Abstract

Generative Engine Optimization (GEO) aims to improve content visibility in AI-generated responses. However, existing methods measure contribution—how much a document influences a response—rather than citation, the mechanism that actually drives traffic back to creators. Also, these methods apply generic rewriting rules uniformly, failing to diagnose why individual documents are not cited. This paper introduces a diagnostic approach to GEO that asks why a document fails to be cited and intervenes accordingly. We develop a unified framework comprising: (1) the first taxonomy of citation failure modes spanning different stages of a citation pipeline; (2) `AgentGEO`, an agentic system that diagnoses failures using this taxonomy, selects targeted repairs from a corresponding tool library, and iterates until citation is achieved; and (3) a document-centric benchmark evaluating whether optimizations generalize across held-out queries. `AgentGEO` achieves 15-40% relative improvement in citation rates across engines and citation methods, while modifying only 5% of content, compared to 25% for baselines. Our analysis reveals that generic optimization can harm long-tail content and some documents face challenges that optimization alone cannot fully address—findings with implications for equitable visibility in AI-mediated information access. Project page: `https://zhihuat.github.io/agentgeo/`.

1 Introduction

The rise of generative engines has sparked significant concern among content creators about traffic loss [1]. Unlike traditional search engines that drive users to click through to websites, generative engines synthesize answers directly, potentially reducing the organic traffic that sustains the creator economy. In response, major platforms have introduced *citation* mechanisms—Google AI Overview displays source links alongside summaries [2], Perplexity provides inline citations [3], and ChatGPT's search mode includes expandable references [4]. These citations are now the primary pathway from synthesized answers back to original webpages.

This pathway, however, is narrow. A recent analysis of 68,879 Google searches reports that only 1% of users who encounter AI summaries click cited sources, compared to 15% who click traditional results when no AI summary appears [5]. Yet the traffic that does arrive could convert at substantially higher rates—one analysis found generative search visitors convert 23 times better than traditional organic visitors [6]. For content creators, citation thus becomes a critical gateway: although click-through is rare, the visitors who do arrive are disproportionately valuable, making the difference between being cited and being overlooked all the more consequential.

From Contribution to Citation. These dynamics have motivated growing interest in Generative Engine Optimization (GEO): modifying web content to improve visibility in AI-generated responses.

*Equal contribution.

†Correspondence to: zhihuat@vt.edu.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Existing work evaluates visibility with *contribution* metrics such as Word Count, which measure the fraction of a response attributed to a source [7]. While useful, these continuous metrics conflate two factors: whether a webpage is cited at all and, conditional on citation, how much of the response is attributed to it. For content creators, the critical barrier is non-citation: no citation means no referral opportunity. We therefore focus on a complementary, citation-first objective that isolates the binary event of being cited. In our experiments, 43% of topically relevant webpages receive no citation under baseline conditions. For these webpages, the question is not "how much am I cited?" but "why am I not cited at all?" Understanding and repairing these failures is an underexplored problem with direct implications for the sustainability of web content creation.

Limitations of Existing Methods. Current GEO methods apply generic preference rules to text uniformly: add statistics, adopt authoritative tone, improve fluency [7, 8]. This approach fails because citation failures are heterogeneous, spanning multiple pipeline stages. A webpage may fail at *fetching* (malformed HTML, JavaScript dependencies), at *parsing* (content truncated or buried below boilerplate), or at *generation* (missing entities, inferior information density versus competitors). Each failure mode requires a different intervention, yet generic rules cannot diagnose which stage failed, nor can they address upstream failures that occur before text is even processed.

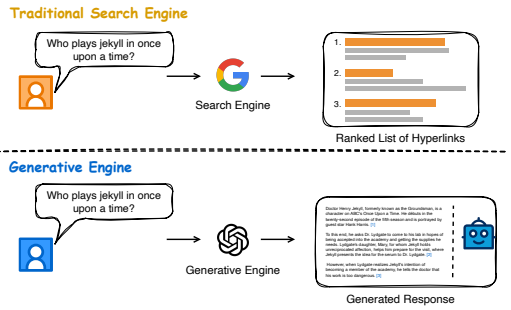


Figure 1: Comparison between traditional search engines and LLM-powered generative engines. Traditional search engines return a ranked list of hyperlinks, while generative engines synthesize answers from retrieved content and provide citations to source webpages.

Contribution 1: A Diagnostic Approach to GEO. Motivated by these observations, we reframe GEO from "what generic improvements might help?" to "why did this webpage fail to be cited for the relevant queries?" We instantiate this perspective in AgentGEO, an agentic framework for iterative diagnosis and repair. Given a webpage that fails to be cited, AgentGEO compares it against cited competitors to identify the most likely disadvantage, selects an appropriate tool from a library of specialized interventions spanning the entire citation pipeline, and verifies whether citation is achieved. If not, it re-diagnoses and iterates, with a memory mechanism to track prior attempts.

Contribution 2: Taxonomy of Citation Failures. We introduce the first systematic taxonomy of citation failure modes spanning the generative engine pipeline: parsing-stage failures (malformed HTML, excessive noise), fetching/context failures (truncation, poor content ordering), and generation-stage failures (entity gaps, intent mismatch, competitor disadvantage). This taxonomy guides our tool library design and provides a foundation for future citation-focused optimization (Figure 2).

Contribution 3: Document-Centric Benchmark. Existing benchmarks pair each document with a single query [7, 8]. In practice, content providers cannot anticipate exact queries; they need optimizations that generalize. We introduce MIMIQ (Multi-Intent Multi-Query), a document-centric benchmark associating each webpage with multiple queries spanning diverse intents, personas, and phrasings. This benchmark enables methods to optimize using a training query set and be evaluated on held-out queries, testing whether optimization produces genuinely more citable content rather than overfitting to specific formulations of a query.

Empirical Takeaways. Our experiments yield fresh insights into why webpages fail to achieve citation and what optimization can—and cannot—fix.

(1) Diagnosis beats generic rules while preserving content integrity (Section 6.2). AgentGEO achieves over 40% relative improvement in citation rates across multiple generative engines. Crucially, it does so with minimal modification: baseline methods alter 25% of original content on average, while our targeted repairs touch only 5%. This suggests that citation failure is rarely a global content quality problem: most webpages need targeted fixes, not extensive rewriting.

(2) Generic rules can harm long-tail content (Section 6.5). We identify topic categories where generic optimization not only fails to improve citation but actively degrades it, whereas diagnostic optimization yields consistent gains. This divergence reflects a fundamental limitation: generic rules

are derived from aggregate patterns, yet specialized topics and underrepresented domains deviate systematically. Diagnostic optimization, by conditioning on each webpage's specific failure mode, avoids this bias and generalizes more equitably across content types.

(3) Not all citation failures are recoverable (Section 6.6). For certain webpages, even diagnostic optimization fails to improve citation. These webpages face disadvantages—dominant competitors—that no content-side modification can overcome. This finding has implications beyond optimization: if some content is systematically disadvantaged regardless of effort, citation mechanisms may amplify certain voices over others, and creator-side optimization alone cannot ensure equitable visibility.

2 Related Work

2.1 Generative Engines

GEs integrate LLMs to retrieve, synthesize, and generate responses directly based on web content utilizing retrieval-augmented generation (RAG) techniques [9, 10]. The workflow of a GE can be roughly divided into three stages: (1) retrieval, (2) content extraction (fetching), and (3) response generation. The retrieval stages identify and retrieve relevant web pages based on user queries. This process may employ traditional search engines [2, 11] or specialized neural retrieval models [10]. The content extraction stage focuses on parsing useful text from the retrieved pages, transforming unstructured web content into structured information suitable for LLM processing. Finally, the response generation stage leverages LLMs to produce coherent, contextually relevant answers derived from the extracted evidence. Recent work further leverages LLM reasoning to iteratively plan and gather evidence for complex queries [12, 13]. These developments board the research focus to the improvement of factual consistency and reliability in responses [14, 15].

2.2 From SEO to GEO

Traditional search engine optimization (SEO) improves webpage ranking through keyword optimization, backlink management, and site structuring [16, 17, 18]. As search shifts toward GEs, GEO has emerged to enhance the visibility of content within generated responses. Existing GEO research has explored two directions of textual modification: rewriting content with manually crafted heuristics [7, 19] or LLM-distilled preference rules [8] to improve legitimate visibility, and crafting adversarial inputs such as prompt injection attacks [20, 21, 22] to manipulate GE behaviors. Despite their differing intents, both lines operate exclusively at the content-level textual modifications. They often neglect critical structural and systemic factors, such as HTML parsing mechanisms, context window constraints, and webpage structure, which significantly influence how content is ingested by the engine. In contrast, `AgentGEO` presents a holistic GEO framework that addresses these limitations by optimizing across multiple aspects of the webpage representation.

2.3 GEO Benchmarks

GEO-Bench [7] is one of the first GEO benchmarks, adopting a query-centric design that pairs each document with a single query and measures visibility gains relative to competing documents after optimization. E-commerce [8] and Researchy-GEO [8] follow the same setting but focus on specific domains. Yet this query-centric setup optimizes and evaluates on the same query, making it impossible to tell whether gains reflect genuinely more citable content or overfitting to that query's specific phrasing. CC-GSEO-Bench [23] adopts a document-centric evaluation, associating each document with multiple queries; yet the number of queries per document is limited and inconsistent, preventing a comprehensive assessment of overall visibility. Moreover, none of these benchmarks evaluate raw HTML structure. Our MIMIQ benchmark (Section 5) addresses both gaps with a document-centric design, and explicit train/test query splits.

3 Understanding Citation Failures

We adopt citation—the binary event of being included as a source—as our optimization target. To understand why documents fail to be cited, we construct a taxonomy of failure modes from contrastive cited–uncited document pairs.

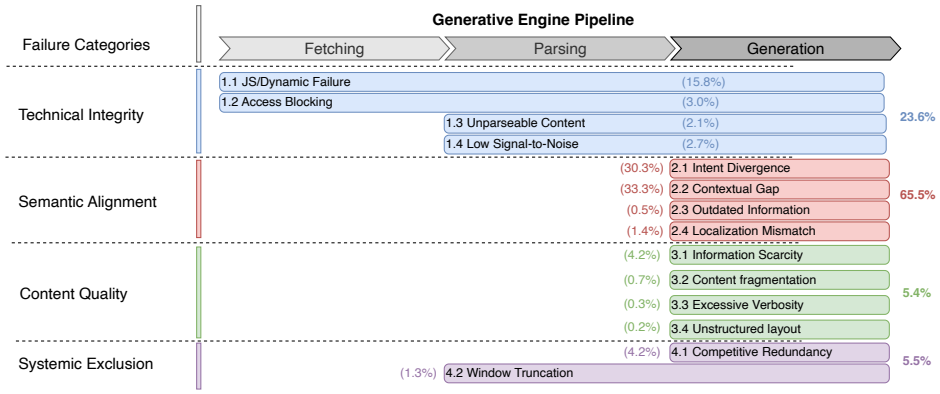


Figure 2: A taxonomy of citation failure modes in generative engines, spanning HTML fetching, parsing and answer generation stages.

3.1 Taxonomy Construction

Data. We construct 949 cited–uncited document pairs from GEO-Bench [7], which spans 25 diverse domains (health, travel, technology, etc.). For each query where a retrieved document was not cited, we pair it with a cited competitor. This pairing isolates the marginal factor: both documents were retrieved for the same query, so the difference lies in why the engine preferred one over the other.

Pipeline. Taxonomy construction proceeded in three stages. (1) Failure pattern extraction: for each pair, an LLM-based diagnostic pipeline compared the two documents against the query and produced a failure diagnosis (prompts in Appendix H.1). (2) Taxonomy design: three annotators (graduate students in NLP) collaboratively grouped recurring failure modes into categories and mapped each to the corresponding pipeline stage, producing the taxonomy in Figure 2. (3) Label validation: three annotators independently re-labeled a random sample of 100 pairs; reaching 84% agreement with LLM labels (Cohen's $\kappa = 0.70$), confirming the reliability of automated labeling at scale.

3.2 The Citation Failure Taxonomy

The taxonomy follows a sequential diagnostic procedure applied to each (query, cited document, uncited document) triple. Each stage corresponds to one failure category, and a document is assigned to the earliest category at which it fails. This design reflects the natural pipeline order—a document must be technically extractable before semantic relevance is assessable, and semantically relevant before content presentation matters. Detailed explanation of each category is provided in Appendix F.

S1: Technical Integrity (23.6%). *Can the document be ingested at all?* Failures occur at the *fetching* stage (JavaScript rendering failures, access blocking) or the *parsing* stage (unparseable content, low signal-to-noise from boilerplate). A webpage with critical information buried in a JavaScript widget or overwhelmed by navigation chrome may never surface to the generator.

S2: Semantic Alignment (65.5%). *Does the extracted content contain information that addresses the query?* This includes intent divergence, contextual gaps (missing specific entities or terminology), outdated information, and localization mismatch. We operationalize this check by extracting the answer evidence from the cited document and verifying whether equivalent information exists in the uncited document. For instance, a query about U.S. tax filing deadlines returns a page on UK self-assessment dates—topically related but for the wrong jurisdiction.

S3: Content Quality (5.4%). *Is the relevant information presented in a form that supports citation?* For documents containing query-relevant information yet remaining uncited, the failure lies in how the information is presented: information scarcity, content fragmentation, excessive verbosity, or unstructured layout. For instance, a document answers a "how does X work" query with three pages of background and a single sentence on the actual mechanism buried near the end—the answer is present, but not in a form the engine surfaces.

S4: Systemic Exclusion (5.5%).*The document passes S1–S3 yet remains uncited; the failure is attributed to factors external to the document itself.* The engine's fixed context window may truncate

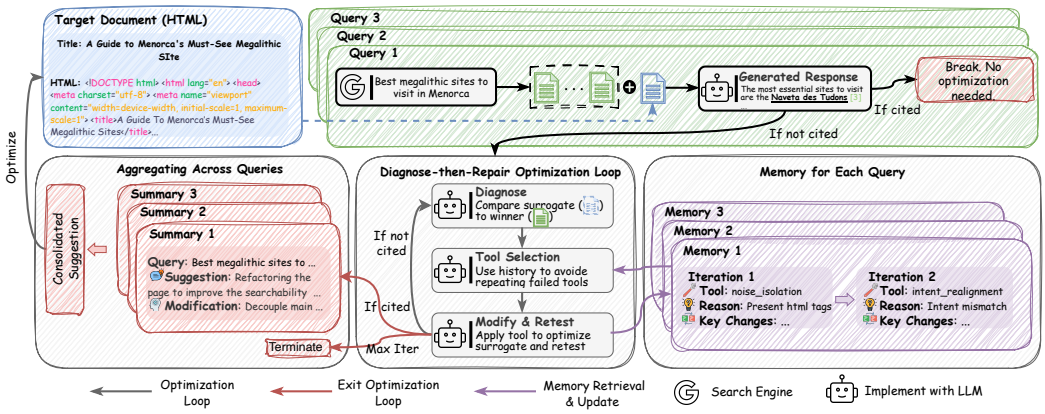


Figure 3: **Optimization process of** `AgentGEO`**.** `AgentGEO` first checks citation status against competitor documents. For uncited ones, it runs a "Diagnose-then-Repair" loop: (1) an LLM diagnoses why the document was not cited; (2) guided by the diagnosis and a query-specific memory of prior iterations, a tool is selected to edit a surrogate copy of the webpage; (3) the surrogate is retested against the same competitors. The loop continues until citation succeeds or a limit is hit. Successful edits are aggregated into a single suggestion applied to the original page, and the procedure iterates over additional training queries for further optimization.

relevant content, or the model may preferentially cite a higher-authority source with the same facts (e.g., Wikipedia). While document-side mitigations can partially help (e.g., content relocation for truncation), the root cause is architectural; we empirically characterize this category in Section 6.6.

4 AgentGEO: A Diagnostic Approach to GEO

Problem Setup. Let d denote a target webpage and $\mathcal{Q}$ a distribution of user queries relevant to d. For query q, let $\mathcal{V}(q, d, \mathcal{G}) \in \{0, 1\}$ indicate whether webpage d is cited by generative engine $\mathcal{G}$. We seek an optimized webpage d^* that maximizes expected citation across $\mathcal{Q}$ while preserving fidelity to the original:

$$d^* = \arg\max_{d'} \mathbb{E}_{q\sim\mathcal{Q}} \left[\mathcal{V}(q, d', \mathcal{G})\right] \quad \text{s.t.} \quad \text{Sim}(d, d') \geq \tau$$

In practice, we optimize using training queries Q_{tr} and evaluate on held-out queries Q_{te}.

Overview. AgentGEO follows a three-stage pipeline (Figure 3). First, for each training query, a *diagnose-then-repair* cycle identifies why the target webpage fails to be cited and applies targeted fixes, guided by the taxonomy and a memory module that tracks prior attempts. Second, a *batch aggregation* stage merges query-specific suggestions into query-invariant modifications, preventing overfitting to individual queries. Third, a *localized editing* mechanism applies modifications at the chunk level, preserving semantic integrity.

4.1 Diagnose-then-Repair Loop

For each training query q, AgentGEO executes an iterative cycle that transforms the target webpage d until it achieves citation or the iteration limit is reached.

Failure Diagnosis. We simulate the GE pipeline: retrieve candidate webpage, generate a response, and record which sources are cited. If the target webpage is not cited, diagnosis begins. To identify the failure reason, the system compares d against the highest-ranked cited competitor and characterizes the vulnerability, i.e., why the competitor was cited instead. These vulnerabilities are classified according to our taxonomy of failure modes (Section 3.2).

Tool Selection with Memory. Given the diagnosed vulnerability, `AgentGEO` selects a repair tool from a library of specialized interventions spanning HTML repair, content reordering, entity enrichment, and information density improvements (full list in Appendix G). To avoid repeating ineffective repairs, we maintain a query-specific memory that tracks prior attempts. At each iteration t, the agent selects a tool based on the current vulnerability and memory. Memory is initialized empty.

Iterative Refinement. Each iteration applies the selected tool to produce a modified webpage $d^{(t)}$ and queries the GE to test whether $d^{(t)}$ is now cited. On success, the system records a *suggestion*—the edit locations and the operations applied—which is later consumed by batch aggregation (Section 4.2). On failure, the system appends the failed attempt to memory and re-runs diagnosis against the current competitors, which may select a different tool in the next iteration. The loop terminates when citation succeeds or the iteration budget is exhausted.

4.2 Aggregating Across Queries

To achieve generalization across training queries, AgentGEO performs webpage updates at the batch level. For each query in a batch, we run the diagnose-then-repair loop (Section 4.1) on an isolated, temporary copy of the current HTML. Each run yields at most one candidate edit, consisting of the diagnosed failure type, a target structural chunk index, and a tool-produced HTML fragment to update that chunk. To keep edits comparable across queries, we fix the chunk partition at the start of the batch and map all candidate edits into this frozen index space, even if the temporary DOM changes during iterative retries. We aggregate candidate edits by target chunk and, for each chunk, select at most one edit to apply to the shared webpage. Our default diagnosis-aware selector ranks candidates by failure severity and a confidence score, resolving conflicts by retaining the highest-ranked proposal. (Other conflict-resolution strategies include heuristic voting within a chunk or an LLM-based arbiter.) The selected per-chunk edits are applied in a single pass to synthesize the batch update, mitigating query-specific overfitting and promoting query- invariant improvements. We do not enforce an explicit similarity threshold at aggregation time; fidelity is primarily maintained through localized chunk-level editing (Section 4.3).

4.3 Localized Editing for faithfulness

To satisfy the semantic preservation constraint, AgentGEO applies modifications at the chunk level rather than rewriting entire webpages. This serves two purposes: it preserves faithfulness by restricting edits to targeted locations, and it avoids the quality degradation LLMs exhibit when processing long webpages [24]. The webpage is partitioned into structure-aware chunks using HTML tags, where each chunk represents a coherent unit (e.g., a paragraph, list, or metadata block). Each suggestion from the aggregation stage specifies the content location it targets; we map these locations to the corresponding chunks. Modifications are applied only to matched chunks; the remaining content and overall HTML structure remain unchanged.

5 Document-Centric Benchmarking

Existing GEO benchmarks pair each document with a single query and evaluate optimization on that same query [7]. This setup risks conflating genuine improvements with query-specific overfitting: an optimization that succeeds for one query may fail for another expressing the same intent.

We introduce **MIMIQ** (**M**ulti-**I**ntent **M**ult**I**-**Q**uery), a document-centric benchmark for evaluating GEO methods. We term it *document-centric* because it treats the document as the unit of optimization and evaluation, mirroring the content creator's setting: they control their document but cannot anticipate the exact queries users will issue. For each document, **MIMIQ** provides multiple queries spanning diverse intents, personas, and phrasings, with explicit train/test splits so that optimizations are evaluated on held-out queries. A detailed comparison to existing benchmarks is provided in Table 6 (Appendix A).

5.1 Construction Pipeline

Webpage Sourcing. We construct a candidate webpage pool from two sources, using each only to obtain webpages; all queries in MIMIQ are subsequently generated from scratch (next paragraph). First, we collect URLs from GEO-Bench [7] query–document pairs and fetch their original HTML. Second, to diversify domain coverage, we issue seed queries to ChatNoir [25] over the ClueWeb22 index [26] and collect the returned pages. Candidates are deduplicated by URL and stratified by topic and content length to prevent skew toward any single domain or short pages. After manual review, we retain 204 webpages spanning 20+ topics with content lengths from a few hundred to over 10,000 words. Sampling heuristics and distribution statistics are provided in Appendix A.

Query Generation. For each webpage, we generate queries through a three-dimensional framework spanning *intent*, *persona*, and *phrasing*. As a preprocessing step, we extract anchor keywords from the page title, as it best reflects the page's topic, organizing them into three categories: core keywords, latent semantic synonyms, and long-tail phrases. We then generate queries by combining each keyword with values sampled along three dimensions: (1) **intent**, drawn from the standard taxonomy of Navigational, Informational, Transactional, and Commercial searches [27, 28], with implausible combinations filtered out (e.g., navigational intent for a recipe page); (2) **persona**, synthesized to capture heterogeneous user backgrounds (e.g., "Domain Expert" vs. "Casual Browser"); and (3) **phrasing**, generating multiple paraphrases per intent–persona combination to capture lexical and syntactic variation. A verifier step removes hallucinations and ensures each query is logically answerable by the target webpage. This yields 60 queries per webpage, partitioned into 20 training and 40 test queries, for a total of 12,240 queries. Prompt templates are provided in Appendix H.2.

5.2 Dataset Variants

Beyond the standard MIMIQ, we build two variants targeting specific optimizer capabilities. Each comprises 50 webpages with 60 queries per page, and follows the same train/test split protocol.

- **MIMIQ-OOD (Out-of-Distribution):** To evaluate generalization to unseen user behaviors, we cluster queries by persona and assign disjoint persona sets to training and test queries. An optimizer that overfits to training-persona phrasings will degrade on held-out personas.
- **MIMIQ-HTML (Structural Robustness):** We curate webpages with complex or noisy DOM structures—cookie notices, navigation residue, JavaScript artifacts, login walls, etc. This variant evaluates whether an optimizer can repair upstream parsing failures, not just rewrite clean text.

6 Experiments

6.1 Experimental Settings

Baselines. We consider three types of baselines. The Vanilla baseline is the original generative engine without any optimization. GEO-Bench [7] proposes eight methods that optimize website content using LLMs. AutoGEO [8] automatically summarizes preference rules from a large number of webpage pairs and optimizes website content accordingly.

Metrics. We evaluate model performance along three dimensions. **Visibility** measures the extent to which the optimized webpage is surfaced by the GE in its response: we report citation rate (CR) — the fraction of test queries on which the optimized webpage is cited — together with Word, Pos, and Wordpos from GEO-Bench [7] to characterize lexical and positional contribution. **Faithfulness** measures the retention of original information after optimization, computed as TF-IDF, Embedding, and Jaccard similarity between the original and optimized content. We further evaluate the **utility** of generated content, including content quality and citation quality in Appendix E. All results are reported as percentages (%). Full metric definitions are in Appendix C.

Generative Engine Setup. We construct a controlled simulation using standard retrieval and generation components with explicit citation instructions. This enables systematic diagnosis, reproducible evaluation, and controlled ablations infeasible with black-box commercial systems. **(1) Retrieval.** We use ChatNoir [25] over the ClueWeb22 [26] corpus (10B webpages) to retrieve the top-5 candidates per query. Since our optimized webpages are not in ChatNoir's index, end-to-end native retrieval is infeasible; we therefore append the target webpage as the final candidate. This isolates the GEO question (*does optimized content get cited more?*) from the orthogonal SEO question (whether the page is retrieved). The placement is also fair by construction: optimized and non-optimized versions of the target are served from the *same* fixed position, so any position-induced bias affects both conditions identically and cancels in the comparison. **(2) Content extraction.** We parse HTML with BeautifulSoup [29] to retain semantic structure. **(3) Generation.** We use GPT-4.1-mini as the default generator under two citation paradigms: inline attribution via in-context generation [30], and pre-hoc attribution via attribute-first-then-generate [31]. These dominate commercial systems (Perplexity [3], ChatGPT Search [4], Claude search [32]).

Table 1: Performance comparison of AgentGEO against baselines across different citation methods. **Bold** and underline indicate the best and second-best results, respectively. "-" indicates the metric is not applicable to the method.

	In-context							Attribute first then generate						
	Citation	Contribution			Faithfulness			Citation	Contribution			Faithfulness		
Method	**CR**	**Word**	**Pos**	**Wordpos**	**TF-IDF**	**Embed**	**Jaccard**	**CR**	**Word**	**Pos**	**Wordpos**	**TF-IDF**	**Embed**	**Jaccard**
Vanilla	56.58	16.28	16.28	16.39	-	-	-	60.20	23.69	28.72	23.57	-	-	-
Technical Terms	55.84	14.78	14.73	14.60	73.32	85.64	21.37	57.33	22.25	27.87	22.03	74.39	86.98	21.98
Cite Sources	60.52	17.01	16.96	16.89	92.76	93.62	65.62	59.21	24.04	29.02	23.79	92.33	92.95	64.60
Keyword Stuffing	62.44	17.83	17.74	17.82	82.85	88.95	30.04	65.62	28.09	33.39	27.81	82.16	86.92	28.39
Unique Words	57.76	15.33	15.37	15.19	86.00	91.58	32.72	57.42	23.43	28.45	23.23	86.11	91.58	33.30
Authoritative	59.14	17.45	17.75	17.57	80.42	86.38	25.15	63.46	26.86	31.85	26.63	80.85	86.35	26.02
Easy-to-Understand	71.11	19.73	19.84	19.75	82.17	86.68	30.26	65.34	28.44	33.62	27.98	81.99	86.93	30.41
Statistics Addition	60.52	17.41	17.29	17.26	95.52	**94.19**	69.31	59.46	24.15	29.39	23.93	95.51	93.75	69.38
Quotation Addition	59.00	16.28	16.16	16.19	**96.78**	93.92	74.85	59.99	23.66	28.88	23.48	96.57	**94.29**	74.78
Fluency Optimization	60.60	16.52	16.63	16.48	92.40	92.72	48.98	62.22	25.44	30.95	25.25	92.20	92.83	49.21
AutoGEO	68.80	17.87	18.53	17.65	67.49	76.99	17.97	65.97	33.96	**34.33**	**32.88**	74.55	78.52	19.77
AgentGEO (ours)	**79.52**	**20.29**	**21.08**	**20.28**	94.23	93.20	**82.40**	**70.00**	**34.18**	31.57	30.92	**97.44**	92.77	**82.66**

Table 2: Performance comparison across different LLM-based generative engines. "-" indicates the metric is not applicable to the method.

	GPT-4.1-mini			GPT-4.1			Claude-haiku-4.5		
Metric	**Vanilla**	**AutoGEO**	**AgentGEO (ours)**	**Vanilla**	**AutoGEO**	**AgentGEO (ours)**	**Vanilla**	**AutoGEO**	**AgentGEO (ours)**
CR	56.58	68.80	**79.52**	76.97	82.43	**84.74**	42.40	46.80	**54.80**
Word	16.28	17.87	**20.29**	28.55	31.67	**32.77**	21.80	24.86	**26.94**
Pos	16.28	18.53	**21.08**	28.70	31.84	**32.53**	21.54	24.81	**27.57**
Wordpos	16.39	17.65	**20.28**	28.68	31.84	**32.35**	21.44	24.66	**26.10**
Precision	48.63	56.79	**58.43**	82.32	**89.25**	82.91	89.42	88.79	**90.33**
Recall	64.62	**68.99**	67.10	88.02	**88.22**	88.07	**96.16**	95.51	96.02
Clarity	61.23	**66.08**	62.88	74.62	**74.88**	72.95	73.99	**75.74**	74.99
Insightfulness	40.20	**45.25**	42.68	64.52	**64.58**	61.56	57.47	**61.05**	60.76
TF-IDF	–	67.49	**94.23**	–	73.56	**94.29**	–	74.27	**95.26**
Embed	–	76.99	**93.20**	–	67.97	**93.85**	–	79.98	**88.76**
Jaccard	–	17.97	**82.40**	–	19.56	**69.26**	–	22.28	**74.58**

6.2 Overall GEO Performance

Performance across different citation methods. We compare AgentGEO with two baseline citation methods: in-context generation [30] and attribute-first-then-generate [31]. In in-context generation, the generative engine is directly prompted to produce answers with citations based on the optimized webpage. In attribute-first-then-generate, it first extracts citation-related attributes from the webpage and then generates answers from them. Table 1 presents the results. AgentGEO consistently outperforms all baselines across both citation methods and generative engines, demonstrating its effectiveness and robustness. In in-context generation, AgentGEO achieves a citation rate of 79.52%, outperforming the best baseline, AutoGEO, by 10.72. In attribute-first-then-generate, it attains 70.00%, surpassing AutoGEO by 4.03. AgentGEO also excels in contribution and faithfulness metrics, improving citation rates while maintaining high-quality, faithful content.

Performance across different LLM backbones. To assess whether AgentGEO's advantages generalize across generators, we evaluate 50 webpages sampled from the standard dataset using three LLMs: GPT-4.1-mini, GPT-4.1, and Claude-haiku-4.5. As shown in Table 2, AgentGEO consistently achieves the highest citation rate, outperforming Vanilla by 22.94, 7.77, and 12.40 percentage points, respectively. Claude generally produces higher-quality responses across all methods, with higher Precision, Recall, Clarity, and Insightfulness than GPT-4.1-mini, suggesting stronger answer synthesis capabilities regardless of optimization. Within each generator, all methods achieve comparable utility scores, indicating that AgentGEO's citation gains do not compromise response quality. Moreover, AgentGEO consistently achieves higher faithfulness than AutoGEO across all generators. On GPT-4.1-mini, its Jaccard similarity reaches 82.40 compared with AutoGEO's 17.97, demonstrating that diagnosis-guided optimization better preserves original content than rule-based rewriting.

Performance under OOD and HTML-compatible settings. Table 3 reports the results. On MIMIQ-OOD, AgentGEO attains a 70.62% citation rate, exceeding AutoGEO by 6.44 points, and ranks first

Table 3: Performance comparison in OOD and HTML compatibility settings.

	MIMIQ-OOD							MIMIQ-HTML						
	Citation	Contribution			Faithfulness			Citation	Contribution			Faithfulness		
Method	**CR**	**Word**	**Pos**	**Wordpos**	**TF-IDF**	**Embed**	**Jaccard**	**CR**	**Word**	**Pos**	**Wordpos**	**TF-IDF**	**Embed**	**Jaccard**
Vanilla	57.14	**25.19**	**25.01**	**24.94**	-	-	-	38.14	18.20	18.41	18.20	-	-	-
AutoGEO	64.18	20.19	20.86	20.66	73.09	79.43	19.48	45.96	23.79	24.12	23.81	65.77	81.51	17.39
AgentGEO (ours)	**70.62**	24.57	24.78	24.42	**96.43**	**92.27**	**83.28**	**67.19**	**32.10**	**32.43**	**32.15**	**82.20**	**95.04**	**63.33**

on all three faithfulness metrics. We attribute this margin to information preservation: AgentGEO modifies only 3.6% of each page's content (TF-IDF), targeting the text responsible for citation failure while leaving the rest intact to support other queries; AutoGEO instead regenerates the entire webpage as a single LLM rewrite, modifying 26.9% of TF-IDF content—roughly 7× more—and discarding information unrelated to the rewriting query but useful for OOD queries at evaluation time. The advantage is amplified on MIMIQ-HTML, where AgentGEO reaches a 67.19% citation rate—a 21.2-point margin over AutoGEO—and achieves the best score on every contribution and faithfulness metric. This gap confirms that HTML structure materially affects citation outcomes and cannot be handled by treating webpages as plain text; AgentGEO addresses this through dedicated HTML-aware editing tools that operate on the page's structural elements rather than discarding them, while AutoGEO's text-level rewrite loses this information.

6.3 Experiments on a Commercial Engine

To assess practical applicability, we evaluate AgentGEO on a commercial search engine. We query ChatGPT in web-search mode—the same mode used in its commercial deployment—across a range of queries, and collect two types of webpages: those cited in the response, and those retrieved but not cited. We optimize the latter on our simulated generative engine, then re-query ChatGPT with the optimized pages included in the candidate set. Experiments are conducted with GPT-5-mini over 160 test queries. Since optimized pages are not in ChatGPT's index, we follow [20] and host them at public URLs, explicitly instructing ChatGPT to fetch them. This preserves the engine's native fetching and parsing pipeline. Both optimized and original versions of each webpage are evaluated under the identical instruction protocol. AgentGEO achieves a citation rate of 48%, compared with 39% for Vanilla—a 23% relative improvement. While it is slightly below Vanilla on Word and Position, it surpasses on WordPos. This confirms that our optimizations can transfer across a commercial engine.

Table 4: Experiments on commercial engine

Method	**Citation Rate**	**Word**	**Position**	**Wordpos**
Vanilla	39.34	**14.19**	**11.24**	14.03
AgentGEO	**48.15**	14.11	9.61	**14.09**

6.4 Experiments in an End-to-End Retrieval Setting

Our previous experiments include the target webpage as the last candidate provided to the generator. While this allows us to evaluate citation performance when the target is included, it does not capture how webpage modifications affect whether the target is retrieved. To examine this effect, we construct a search index containing the target webpage and competing documents, and rank them using BM25 [33], a widely adopted lexical ranking function. Only the top-K retrieved documents are passed to the generator, allowing webpage modifications to affect both retrieval and subsequent citation. For each target, we construct a competitor pool of size N and pass the top-K retrieved candidates to the generator. We measure visibility, citation conditional on visibility, and end-to-end citation, where $P(\text{cited}) = P(\text{visible}) \cdot P(\text{cited} \mid \text{visible})$. As shown in Table 5, AgentGEO improves end-to-end citation over Vanilla by 14.5–16.2 percentage points across all four settings. The gains reflect both increased retrieval visibility and higher conditional citation rates. For example, at $N = 50$ and $K = 10$, visibility increases from 51.7% to 67.4%, while conditional citation rises from 85.2% to 89.4%, improving end-to-end citation from 43.6% to 59.8%. These results demonstrate that AgentGEO remains effective when target exposure depends on organic retrieval rather than guaranteed inclusion.

Table 5: Retrieval visibility and citation performance under BM25 retrieval. N is the competitor pool size, and K is the number of retrieved documents passed to the generator.

N	K	Visible (%)		Cite|Vis (%)		Cited (%)	
		Vanilla	AgentGEO	Vanilla	AgentGEO	Vanilla	AgentGEO
30	10	68.2	**80.2**	84.8	**91.6**	57.3	**73.0**
50	10	51.7	**67.4**	85.2	**89.4**	43.6	**59.8**
30	5	47.2	**60.6**	91.1	**94.7**	42.6	**57.1**
50	5	33.5	**47.7**	90.0	**95.6**	29.9	**45.3**

6.5 Evaluation of Different Topics

Figure 4 breaks down citation rate improvement over the vanilla baseline by topic (arts, history, etc.). `AgentGEO` yields positive improvements across nearly all topics, with the largest gains on arts and entertainment, history, and politics. AutoGEO, in contrast, drops below the vanilla baseline on 5 out of 10 topics—particularly those with already high original citation rates—indicating that a single rewriting rule applied uniformly cannot accommodate topic-specific failure modes and risks degrading content that did not need fixing. `AgentGEO` avoids this regression because each webpage is diagnosed individually, and edits to one document do not constrain edits to another. The exception is the health topic, where the vanilla citation rate is already near 80%, leaving little room for improvement.

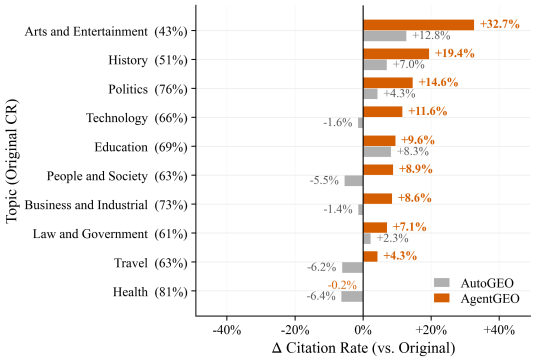


Figure 4: Citation rate change over vanilla baseline across different topics. `AgentGEO` improves 9 of 10 topics while AutoGEO degrades in 5 of 10, particularly those with already-high baseline rates.

6.6 Failure Optimization Case Study

To probe the boundaries of `AgentGEO`, we examine the failure cases from our main optimization run: although the overall citation rate rises from 57.0% to 83.7% on training queries, 163 queries still fail to elicit a citation of their target webpage. Inspecting these cases reveals a consistent pattern: the optimization layer is not the bottleneck. The pages have, in most cases, been successfully edited to address the failure mode `AgentGEO` diagnosed for them, yet the GE still elects to cite a different source. The failures lie not in our pipeline's ability to act on a diagnosis, but in the GE's preference for sources whose advantage is not reducible to page content. Consider a university course page on machine learning targeted by the query *"best online machine learning courses"*. `AgentGEO` expands the page with detailed course descriptions, prerequisites, and learning outcomes, addressing the diagnosed coverage gap. The GE nevertheless cites Coursera in its place, presumably reflecting source-level priors—platform reputation, domain authority—that operate independently of page content. Such priors form a hard ceiling for any purely content-based optimization, and call for deeper investigation into the systemic factors underlying GE citation mechanisms.

7 Conclusion

This paper presents a diagnostic approach to Generative Engine Optimization that shifts focus from generic content rewriting to understanding why individual webpages fail to be cited. We introduce a taxonomy of citation failure modes, `AgentGEO` —an agentic system that diagnoses failures and applies targeted repairs—and a webpage-centric benchmark for comprehensive evaluation. Future work could focus on validating `AgentGEO` against commercial engines and improving scalability through distillation and efficient diagnosis strategies.

Acknowledgments and Disclosure of Funding

We acknowledge support from the National Science Foundation through grants IIS-2312794, IIS-2313130, OAC-2239622, CNS-2424127, OAC-2613761, and the Amazon–Virginia Tech Initiative for Efficient and Robust Machine Learning.

References

[1] The Economist. Ai is killing the web: Can anything save it? *The Economist*, July 2025. Accessed: 2026-01-28.

[2] Google. Ai overviews and ai mode in search. White paper, Google, May 2025.

[3] Perplexity. Perplexity ai, 2026. Accessed: 2026-01-20.

[4] OpenAI. Introducing ChatGPT search, October 31 2024.

[5] Athena Chapekis and Anna Lieb. Google users are less likely to click on links when an ai summary appears in the results. Pew Research Center, July 22 2025. Accessed: 2026-01-28.

[6] Patrick Stox. Does AI search traffic convert better than traditional search? for Ahrefs, yes: 0.5% of visitors drove 12.1% of signups. Ahrefs Blog, June 16 2025. Accessed: 2026-01-28.

[7] Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, and Ameet Deshpande. Geo: Generative engine optimization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5–16, 2024.

[8] Yujiang Wu, Shanshan Zhong, Yubin Kim, and Chenyan Xiong. What generative search engines like and how to optimize web content cooperatively. *arXiv preprint arXiv:2510.11438*, 2025.

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

[10] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781, 2020.

[11] Simon Willison. Anthropic trust center: Brave search added as a subprocessor. `https://simonwillison.net/2025/Mar/21/anthropic-use-brave/`, March 2025.

[12] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

[13] Fengran Mo, Chuan Meng, Mohammad Aliannejadi, and Jian-Yun Nie. Conversational search: From fundamentals to frontiers in the llm era. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4094–4097, 2025.

[14] Alireza Salemi and Hamed Zamani. Evaluating retrieval quality in retrieval-augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2395–2400, 2024.

[15] Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878, 2024.

[16] Harold Davis. *Search engine optimization*. " O'Reilly Media, Inc.", 2006.

[17] Nursel Yalçın and Utku Köse. What is search engine optimization: Seo? *Procedia-Social and Behavioral Sciences*, 9:487–493, 2010.

[18] John B Killoran. How to use search engine optimization techniques to increase website visibility. *IEEE Transactions on professional communication*, 56(1):50–66, 2013.

[19] Mahe Chen, Xiaoxuan Wang, Kaiwen Chen, and Nick Koudas. Generative engine optimization: How to dominate ai search. *arXiv preprint arXiv:2509.08919*, 2025.

[20] Fredrik Nestaas, Edoardo Debenedetti, and Florian Tramèr. Adversarial search engine optimization for large language models. *arXiv preprint arXiv:2406.18382*, 2024.

[21] Samuel Pfrommer, Yatong Bai, Tanmay Gautam, and Somayeh Sojoudi. Ranking manipulation for conversational search engines. *arXiv preprint arXiv:2406.03589*, 2024.

[22] Yiming Tang, Yi Fan, Chenxiao Yu, Tiankai Yang, Yue Zhao, and Xiyang Hu. Stealthrank: Llm ranking manipulation via stealthy prompt optimization. *arXiv preprint arXiv:2504.05804*, 2025.

[23] Qiyuan Chen, Jiahe Chen, Hongsen Huang, Qian Shao, Jintai Chen, Renjie Hua, Hongxia Xu, Ruijia Wu, Ren Chuan, and Jian Wu. Cc-gseo-bench: A content-centric benchmark for measuring source influence in generative search engines, 2025.

[24] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

[25] Janek Bevendorff, Benno Stein, Matthias Hagen, and Martin Potthast. Elastic ChatNoir: Search Engine for the ClueWeb and the Common Crawl. In Leif Azzopardi, Allan Hanbury, Gabriella Pasi, and Benjamin Piwowarski, editors, *Advances in Information Retrieval. 40th European Conference on IR Research (ECIR 2018)*, Lecture Notes in Computer Science, Berlin Heidelberg New York, March 2018. Springer.

[26] Arnold Overwijk, Chenyan Xiong, and Jamie Callan. Clueweb22: 10 billion web documents with rich information. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 3360–3362, 2022.

[27] Andrei Broder. A taxonomy of web search. In *ACM Sigir forum*, volume 36, pages 3–10. ACM New York, NY, USA, 2002.

[28] Daniel E Rose and Danny Levinson. Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*, pages 13–19, 2004.

[29] Leonard Richardson. Beautiful soup. `https://www.crummy.com/software/BeautifulSoup/`, 2025.

[30] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*, 2023.

[31] Aviv Slobodkin, Eran Hirsch, Arie Cattan, Tal Schuster, and Ido Dagan. Attribute first, then generate: Locally-attributable grounded text generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3344, 2024.

[32] Anthropic. Citations - Anthropic Claude Documentation. `https://platform.claude.com/docs/en/build-with-claude/citations`, 2024. Accessed: 2026-04-20.

[33] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and trends® in information retrieval*, 4(1-2):1–174, 2009.

[34] Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, et al. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*, 2022.

[35] João Coelho, Jingjie Ning, Jingyuan He, Kangrui Mao, Abhijay Paladugu, Pranav Setlur, Jiahe Jin, Jamie Callan, João Magalhães, Bruno Martins, et al. Deepresearchgym: A free, transparent, and reproducible evaluation sandbox for deep research. *arXiv preprint arXiv:2505.19253*, 2025.

[36] qx labs. agents-deep-research, 2026.

Appendix Overview

This appendix provides additional details on benchmark construction, evaluation, experiments, and the implementation of AgentGEO.

Section	Contents
A. Benchmark Construction	Data collection, sampling, and benchmark variants
B. Citation Mechanisms	Citation paradigms in generative engines
C. Evaluation Metrics	Citation, faithfulness, and utility metrics
D. Cost Analysis	LLM calls and optimization cost
E. Additional Experiments	Cross-model transfer, robustness, and ablations
F. Citation Failure Taxonomy	Failure-mode definitions and examples
G. Optimization Tools	Repair operators and tool-selection policy
H. Prompt Templates	Prompts used throughout the framework

Broader Impacts

This research introduces `AgentGEO`, a framework for Generative Engine Optimization (GEO). While GEO empowers content providers to remain visible in an evolving search landscape, it also introduces several potential risks that warrant ethical consideration:

- Information Integrity and Manipulation: The ability to systematically optimize content for LLM citations could be exploited to prioritize biased, low-quality, or even misleading information. If malicious actors use GEO to ensure their content is cited more frequently than authoritative sources, the reliability of generative engines as a primary information gatekeeper could be severely compromised.
- Degradation of Human-Centric Content: Similar to the historical "SEO-spam" in traditional search, a widespread adoption of GEO may incentivize creators to write for "engines" rather than humans. This could lead to a web ecosystem filled with "LLM-friendly" but redundant or stylistically unnatural content, ultimately degrading the quality of information available to human readers.

A Benchmark Construction

We construct the MIMIQ benchmarks using real-world web data. Specifically, we leverage GEO-Bench and ChatNoir to build a candidate pool of web pages, from which we sample and filter to curate the final dataset. GEO-Bench provides query-document pairs, each consisting of a query and five candidate documents. We use the associated URLs to retrieve the original HTML pages.

To diversify the data sources, we supplement the dataset using ChatNoir's search [25] over the ClueWeb22 index [26]. Specifically, we issue 40 diverse queries spanning 10 broad domains (science, health, technology, history, business, education, environment, law, sports, and arts), each returning up to 50 results. Retrieved documents are deduplicated against the GEO-Bench candidates by URL.

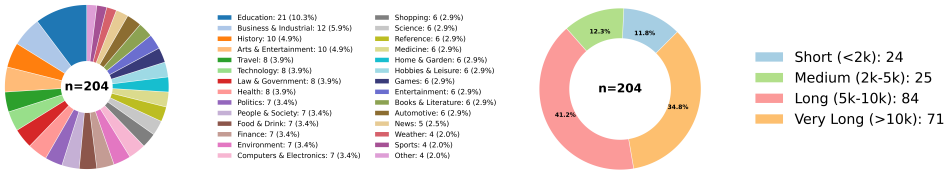
MIMIQ. We categorize each document into four length groups by words count: Short ($<$2K), Medium (2K–5K), Long (5K–10K), and Very Long ($>$10K). Based on length group and topic, we apply a priority-based sampling strategy: (1) at least 100 documents must belong to the Medium, Long, or Very Long groups to prevent domination by short web pages; (2) at least 20 distinct topics must be represented, prioritizing 15 high-importance topics (e.g., science, health, technology, history); and (3) remaining slots are distributed evenly across topics to maintain balance. This yields 210 candidates, which after manual review produce the final MIMIQ benchmark of 204 documents.

Figure 5 presents the distribution of topics and content lengths within the MIMIQ benchmark. As illustrated in Figure 5a, the benchmark spans a wide range of subjects, including technology, health, finance, education, and entertainment, ensuring comprehensive coverage of various domains. Figure 5b shows that webpage lengths vary significantly, ranging from brief articles of a few hundred words to extensive reports exceeding 10,000 words. This diversity in both thematic content and

Table 6: Comparison of **MIMIQ** and existing GEO benchmarks. Our dataset combines a document-centric paradigm with a holistic evaluation scope and supports explicit train/test query splits.

Dataset	Doc-centric	HTML	Train/Test Split	Queries/doc
GEO-Bench [7]/E-commerce [8]/Researchy-GEO [8]	✗	✗	✓	1
CC-GSEO-Bench [23]	✓	✗	✗	2-10
MIMIQ (Ours)	✓	✓	✓	60

webpage size enhances the benchmark's utility for evaluating information retrieval and citation systems.



(a) Topic Distribution (b) Content Length Distribution

Figure 5: Distribution of topics and content lengths in the MIMIQ benchmark.

MIMIQ-OOD. MIMIQ-OOD is constructed by randomly sampling 50 documents from MIMIQ. The key difference lies in the train/test query split. Recall that we simulate diverse search intents by generating queries from different user personas (cf. Section 5); in MIMIQ-OOD, we assign disjoint sets of personas to generate training and test queries, enabling evaluation of generalization to unseen query distributions.

MIMIQ-HTML. From the same candidate pool, we curate a subset of 50 documents that exhibit common HTML parsing artifacts, including cookie notices, navigation residue, footer elements, JavaScript code residue, empty or 404 pages, encoding issues, boilerplate text (e.g., "click here"), and login walls. This subset enables targeted evaluation of systems' robustness to noisy, imperfectly parsed web content.

B Citation Mechanisms in Generative Engines

How a generative engine attributes its response to sources determines where citation failures can occur. We identify the following three paradigms.

Inline Attribution. Reference markers (e.g., "[1]") are embedded directly within the generated sequence, providing fine-grained provenance at the sentence or clause level. The model decides *during generation* which sources support each claim [30]. Systems like Perplexity and ChatGPT's search mode employ this approach [3, 4].

Pre-hoc Attribution. The system first selects passages from retrieved documents, then generates a response grounded exclusively in these selections. Because generation operates over pre-selected evidence, the source context serves as an inherent anchor—the model cannot easily hallucinate beyond what was extracted [31]. However, relevant content not selected upstream will not be cited. System like Claude's search mode employ this approach [32].

Post-hoc Attribution. Generation and verification are decoupled: the model first produces a response, then a separate module retrospectively identifies supporting sources [34, 30]. While flexible, this paradigm risks hallucination propagation—errors introduced during generation may persist if the verifier incorrectly validates unsupported claims.

Given their prevalence in commercial systems and tighter grounding guarantees, we focus on inline and pre-hoc attribution. These paradigms expose different failure surfaces: inline attribution can

fail when the model overlooks a relevant source during generation, while pre-hoc attribution can fail earlier, during passage selection. Our taxonomy accounts for failures across both paradigms.

C Metric Definitions

We provide the definition of all metrics used in our experiments. We organize them along three dimensions: visibility, faithfulness, and utility.

C.1 Visibility

Let $\mathcal{Q}_{\text{te}}$ denote the held-out test query set for a target webpage d, and let r_q denote the response generated by the GE for query q with citations to retrieved documents.

Citation Rate (CR). The fraction of test queries on which the target webpage is cited at least once:

$$\text{CR}(d) = \frac{1}{|\mathcal{Q}_{\text{te}}|} \sum_{q \in \mathcal{Q}_{\text{te}}} \mathbb{1}[\, d \in \text{cited}(r_q)\,]. \tag{1}$$

This is our primary metric. CR captures the binary event of being cited, directly reflecting whether a webpage receives any referral opportunity from the GE.

Word, Pos, Wordpos. Following GEO-Bench [7], these metrics assess the lexical and positional share that the cited webpage contributes to the response, conditional on being cited:

- **Word** measures the fraction of words in the response attributable to d, weighted by lexical overlap between cited spans and d's content.
- **Pos** measures the positional prominence of d's contribution, with citations appearing earlier in the response receiving higher weight.
- **Wordpos** combines lexical and positional share into a single score.

Whereas CR captures whether d is cited at all, Word/Pos/Wordpos quantify how much of the response is attributable to d given citation. We treat CR as primary and report Word/Pos/Wordpos as complementary. We refer readers to [7] for the precise formulations.

C.2 Faithfulness

Let d denote the original webpage and d' the optimized version. Faithfulness metrics quantify the preservation of original information after optimization.

TF-IDF Similarity. Cosine similarity between TF-IDF weighted term vectors of d and d':

$$\text{TF-IDF}(d, d') = \cos(\mathbf{v}_d^{\text{tfidf}}, \mathbf{v}_{d'}^{\text{tfidf}}). \tag{2}$$

This captures preservation of term-level content with weighting by term importance, penalizing the loss or alteration of distinctive vocabulary.

Embedding Similarity. Cosine similarity between dense semantic embeddings of d and d', computed using OpenAI's `text-embedding-3-small`:

$$\text{Embed}(d, d') = \cos(\mathbf{e}_d, \mathbf{e}_{d'}). \tag{3}$$

This captures semantic preservation: even if surface words change, content with similar meaning yields high similarity.

Jaccard Similarity. Strict lexical overlap between unique word sets:

$$\text{Jaccard}(d, d') = \frac{|W(d) \cap W(d')|}{|W(d) \cup W(d')|}, \tag{4}$$

where $W(\cdot)$ denotes the set of unique tokens after standard preprocessing. Jaccard is the strictest faithfulness metric: any word added or removed reduces the score, making it a sensitive indicator of how much surface content has been altered.

The three metrics jointly characterize faithfulness from term-weighted, semantic, and strict lexical perspectives.

C.3 Utility

Utility metrics evaluate whether the optimization improves the end product—the GE-generated response—rather than just citation visibility. We decompose utility into content quality and citation quality.

Content Quality. Following DeepResearchGym [35], we evaluate generated responses along two axes:

- **Relevance** measures how well the response addresses the query. We report **Precision** (the fraction of response content that is relevant to the query) and **Recall** (the fraction of query-relevant information present in the response).
- **Quality** measures the readability and substantiveness of the response. We report **Clarity** (coherence and readability, judged by an LLM evaluator) and **Insightfulness** (depth and substantiveness, judged by an LLM evaluator).

Citation Quality. Citation quality measures whether the citations to the optimized webpage genuinely support the generated claims. Following Gao et al. [30], we report:

- **Citation Recall**: the fraction of generated statements that are fully supported by their cited passages.
- **Citation Precision**: the fraction of non-removable citations, i.e., citations whose removal would leave the corresponding statement unsupported.

Together, citation recall measures whether claims have adequate evidentiary support, and citation precision measures whether each citation is necessary rather than redundant. Entailment for both metrics is judged by an LLM evaluator following the protocol of [30].

D Cost Analysis

Each diagnose-then-repair iteration involves four LLM calls (diagnosis, tool selection, repair, and citation re-verification), compared to two calls for both GEO-Bench and AutoGEO (rewriting, and re-verification). With 20 training queries per document and an average of two iterations per query, AgentGEO requires approximately 160 LLM calls per document, costing around $0.49 using GPT-4.1-mini. This overhead is justified by AgentGEO's substantially higher citation rate and content preservation, particularly given that the cost remains modest in absolute terms.

E Additional Experiments

This section presents additional experiments on generalization (E.1–E.3), content and optimization analysis (E.4–E.7), and red-teaming and ablation studies (E.8–E.9).

E.1 Performance on Deep Research Systems

Deep research systems are analogous to generative engines: both retrieve web content and synthesize cited responses. Unlike single-pass GEs, however, deep research systems iteratively identify knowledge gaps and issue fine-grained follow-up searches before producing a final answer. This iterative behavior makes them a meaningful alternative evaluation environment for testing whether GEO methods transfer beyond single-pass settings. We evaluate on a deep research system implemented with the open-source `qx-labs/agents-deep-research` framework [36]. Table 7 reports the results. AgentGEO improves citation rate from 78.00% to 84.47%, an absolute gain of 6.47 points over Vanilla, and substantially improves all contribution metrics, with Wordpos increasing from 27.65 to 34.31, a 24.1% relative gain. Notably, AutoGEO *decreases* citation rate to 69.00% in this setting, despite achieving comparable contribution scores to Vanilla. This pattern echoes our finding in Section 6.5: generic rewriting rules can hurt visibility when the target system has its own preferences, whereas diagnostic optimization remains robust. Content quality metrics are comparable across all methods, indicating that AgentGEO's citation gains do not come at the cost of response quality. These results suggest that AgentGEO's diagnostic approach generalizes beyond standard GE pipelines to iterative retrieval-synthesis systems.

Table 7: Performance on a deep research system.

	Citation	Contribution			Content Quality			
Method	**CR**	**Word**	**Pos**	**Wordpos**	**Precision**	**Recall**	**Clarity**	**Insight**
Vanilla	78.00	28.11	27.39	27.65	**11.07**	86.43	68.72	47.40
AutoGEO	69.00	27.82	28.26	28.42	9.22	**86.84**	**70.80**	**50.45**
AgentGEO (ours)	**84.47**	**34.60**	**34.00**	**34.31**	9.86	85.92	68.15	48.59

Table 8: Cross-Citation Mechanism Robustness Evaluation.

	Citation	Contribution			Faithfulness		
Method	CR	Word	Pos	Wordpos	TF-IDF	Embed	Jaccard
Vanilla	69.98	26.09	25.98	25.93	–	–	–
`AgentGEO`	**84.29**	**35.18**	**34.89**	**34.54**	94.91	90.32	75.36
Improv.	*+14.31*	*+9.09*	*+8.91*	*+8.61*	–	–	–

E.2 Cross-Citation Mechanism Robustness Analysis

To assess the generalization capability of `AgentGEO` across different citation mechanisms, we conducted a cross-method robustness experiment on a subset of 50 webpages. We enforced a distribution shift between optimization and evaluation: `AgentGEO` was optimized using *Attribute-First* feedback but evaluated under the *In-Context* generation paradigm. This setup strictly tests whether structural repairs translate across distinct engine architectures. As shown in Table 8, `AgentGEO` demonstrates strong transferability, improving the Citation Rate by **14.31** (from 69.98% to 84.29%) and boosting Contribution metrics by over 9%. Coupled with high faithfulness scores (TF-IDF: 94.91%, Embedding: 90.32%), these results confirm that `AgentGEO` addresses fundamental content deficiencies rather than overfitting to specific engine artifacts.

E.3 Cross-Model Transfer of AgentGEO Optimizations

To evaluate whether webpages optimized by `AgentGEO` transfer to different LLM families, we optimize webpages using GPT-4.1-mini and evaluate the original and optimized versions with two other generators: DeepSeek-V4-Flash and Kimi-K2.6. As shown in Table 9, the citation rate increases from 69.20 to 76.74 on DeepSeek-V4-Flash and from 71.47 to 78.74 on Kimi-K2.6. The Word, Pos, and WordPos metrics also improve on both generators. These results indicate that the improvements obtained using GPT-4.1-mini also transfer to the two evaluated model families.

E.4 Evaluation of Different Content Length

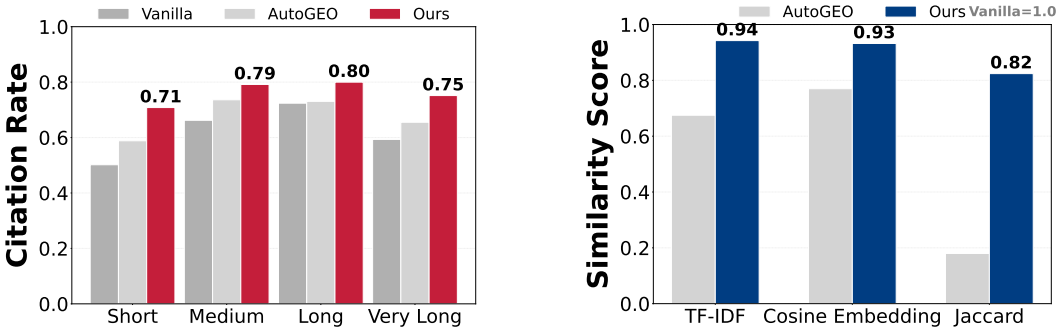
We analyze the performance of `AgentGEO` across webpages of varying lengths. We categorize the webpages into four groups based on their length: short ($< 2k$ words), medium (2k-5k words), long (5k-10k words), and very long ($>$ 10k words). We then evaluate the citation rate improvement over the vanilla baseline for each length category. The results are summarized in Figure 6. The results indicate that `AgentGEO` consistently improves citation rates across all length categories, with the most significant improvements observed in very long webpages. This suggests that chunking-based constrained editing method is particularly effective at optimizing webpages that have sufficient content to benefit from structural and informational enhancements, while still being manageable in size for the optimization process.

E.5 Accuracy of Optimization-Time Diagnosis and Repair Selection

To evaluate the accuracy of `AgentGEO`'s diagnoses during optimization, we randomly sampled 200 logged diagnoses for manual evaluation. Human annotators independently labeled each case based on the target webpage, competing webpage, and query. As shown in Table 10, `AgentGEO` correctly identifies the issue category in 185 out of 200 cases, achieving 92.5% accuracy and a macro-F1 score of 0.909. Among the 15 misclassified cases, 11 were incorrectly classified as Systemic Exclusion.

Table 9: Cross-model transfer of AgentGEO optimizations. Webpages are optimized using GPT-4.1-mini and evaluated with two other generators.

Model	Method	Citation	Contribution		
		CR	Word	Pos	WordPos
DeepSeek-V4-Flash	Original	69.20	25.44	24.91	24.97
	`AgentGEO`	**76.74**	**30.38**	**29.48**	**29.68**
	Improv.	*+7.54*	*+4.94*	*+4.57*	*+4.71*
Kimi-K2.6	Original	71.47	21.89	22.19	21.85
	`AgentGEO`	**78.74**	**26.68**	**27.20**	**26.59**
	Improv.	*+7.27*	*+4.79*	*+5.01*	*+4.74*



(a) Citation rate across different content lengths

(b) Content faithfulness (Similarity Scores)

Figure 6: Performance analysis on webpage constraints. (a) Comparison of citation rates across varying webpage lengths. (b) Evaluation of content preservation using similarity metrics (TF-IDF, Cosine Embedding, and Jaccard).

We further annotated the appropriate repair tool for each of the 200 cases. `AgentGEO` selects the annotated tool in 163 cases (81.5% accuracy), achieving a macro-F1 score of 0.855. As shown in Table 11, `structure_optimization` and `data_serialization` achieve perfect precision and recall on the audited sample. In contrast, `bluf_optimization` achieves perfect recall but lower precision (0.562), indicating that it is also selected for cases assigned to other tools. This pattern is consistent with the fallback mechanism described in Appendix G.2.3, which may select `bluf_optimization` when earlier repair attempts fail. A similar pattern appears for `content_relocation`, with precision of 0.747 and recall of 0.954.

E.6 Citation Quality

Beyond whether a webpage is cited, we evaluate whether the resulting citations genuinely support the generated claims. We evaluate 20 documents $\times$ 20 test queries (400 query–document pairs in total), reporting citation recall and citation precision.

Table 12 reports the results. `AgentGEO` achieves a citation recall of 47.73% and a citation precision of 69.30%, surpassing Vanilla by 11.88 and 8.23 points, respectively. In contrast, AutoGEO (35.83% / 60.74%) is essentially indistinguishable from Vanilla (35.85% / 61.07%) on both metrics, despite modifying 25% of the original content. `AgentGEO` attains its gains while modifying only 5% of content (Section 6). This contrast suggests that `AgentGEO`'s targeted modifications jointly improve the likelihood of being cited (as reflected by the higher CR in Section 6) and the grounding quality of the resulting citations. We note that the absolute citation quality scores are consistent with those reported in the original ALCE evaluation [30], reflecting the inherent difficulty of fine-grained citation attribution.

Table 10: Accuracy of optimization-time diagnoses on 200 manually annotated cases. TI, SA, CQ, and SE denote Technical Integrity, Semantic Alignment, Content Quality, and Systemic Exclusion, respectively.

	AgentGEO Prediction				Performance		
Human Label	TI	SA	CQ	SE	Precision	Recall	F1
Technical Integrity	**17**	0	1	5	1.000	0.739	0.850
Semantic Alignment	0	**69**	2	5	0.986	0.908	0.945
Content Quality	0	1	**26**	1	0.897	0.929	0.912
Systemic Exclusion	0	0	0	**73**	0.869	1.000	0.930

Table 11: Accuracy of repair tool selection on 200 manually annotated cases.

Repair Tool	Precision	Recall	F1
`structure_optimization`	**1.000**	**1.000**	**1.000**
`data_serialization`	**1.000**	**1.000**	**1.000**
`intent_realignment`	0.900	0.818	0.857
`content_relocation`	0.747	0.954	0.838
`entity_injection`	0.971	0.710	0.820
`noise_isolation`	0.667	0.857	0.750
`bluf_optimization`	0.562	**1.000**	0.720

E.7 Scaling Properties of Training Data

To evaluate scalability, we vary the training set size from 0 to 80 queries while keeping a fixed test set of 20 queries. As shown in Table 13, citation performance improves consistently with training volume, demonstrating a clear positive scaling trend. The mean citation rate increases steadily with training size, reaching a 17.65% relative improvement at 80 queries. Concurrently, the standard deviation decreases consistently from 0.2050 to 0.1625, indicating that larger training sets enhance not only citation accuracy but also stability.

E.8 Red-Teaming Analysis

To examine whether `AgentGEO` increases the citation and propagation of misleading content, we construct 13 adversarial webpages: nine containing modified factual values and four containing fabricated statistics attributed to nonexistent institutions. Each webpage is paired with 10 queries, yielding 130 evaluation cases. We optimize these webpages using `AgentGEO` and compare the citation rate and false-claim propagation before and after optimization.

As shown in Table 14, `AgentGEO` increases the citation rate of webpages containing false claims from 73.85% to 85.38%, an improvement of 11.53 percentage points. However, the number of generated answers reproducing a false claim remains unchanged at six. Consequently, the false-claim propagation rate among cited answers decreases from 6.25% to 5.41%. Similarly, among answers addressing the modified fact, the propagation rate decreases from 19.35% to 16.67%. The number of answers addressing the modified fact increases from 31 to 36, with the five additional answers reporting the correct value.

These results show that `AgentGEO` can increase citations to webpages containing false information without increasing the observed number of false claims reproduced in generated answers. One possible explanation is that optimization improves other relevant content on these webpages, leading to additional citations without reinforcing the false claims. Nevertheless, increased citation of misleading webpages remains a potential concern, even when their false claims are not reproduced. This distinction highlights the importance of evaluating webpage citation and claim propagation separately.

Table 12: Citation quality evaluation. AgentGEO improves both citation recall and precision, while AutoGEO shows no meaningful improvement over Vanilla.

Method	Citation Recall	Citation Precision
Vanilla	35.85	61.07
AutoGEO	35.83	60.74
AgentGEO **(ours)**	**47.73**	**69.30**

Table 13: Citation Performance Across Training Set Sizes

# Training Queries	Mean	Std. Dev.	Abs. Improv.	Rel. Improv.
0	0.6837	0.2050	–	–
10	0.7200	0.2112	0.0363	5.30%
20	0.7214	0.1994	0.0377	5.51%
40	0.7614	0.1910	0.0777	11.36%
60	0.7729	0.1767	0.0891	13.04%
80	0.8044	0.1625	0.1207	17.65%

E.9 Ablation Study

Memory-based Tool Selection. To evaluate the effectiveness of the memory-based tool selection mechanism in AgentGEO, we conduct an ablation study by comparing the performance with and without this component. The results are presented in Table 15.

With the memory module, the citation rate reaches 79.32%, surpassing the variant without this module by 3.38. Meanwhile, across the three contribution metrics (Word/Pos/Wordpos), the memory-equipped variant consistently outperforms its counterpart, with improvements of 0.48, 0.38, and 0.44, respectively. This indicates that memory enables the optimizer to select more effective tools based on prior edit outcomes, rather than repeatedly invoking the same tool without informed guidance. Faithfulness scores also favor the memory-equipped variant across all three metrics (TF-IDF: $+0.06$, Embed: $+1.06$, Jaccard: $+0.35$), suggesting that memory not only improves citation effectiveness but also helps preserve semantic consistency with the original content by avoiding redundant or conflicting edits. Together, these consistent gains across all metric groups demonstrate that the memory module is a necessary component for stable and well-rounded optimization.

Batch Size. To investigate the impact of batch size on the performance of AgentGEO, we conduct experiments with different batch sizes, including 1 (sequential modification), 5, 10, and 20. The results are presented in Table 15.

With a batch size of 5, the citation rate reaches 80.86%, surpassing the variants with batch sizes of 1, 10, and 20 by 6.33%, 5.02%, and 5.48%, respectively. However, on the three contribution metrics (Word/Pos/Wordpos), a batch size of 1 achieves the highest scores (25.56, 25.57, 25.52), exceeding batch size 5 by 2.17, 2.17, and 2.29. This indicates that smaller batches enable more aggressive query-specific edits, while moderate batching better captures cross-query optimization patterns that translate into citation gains. For faithfulness, a batch size of 1 yields the lowest scores across all three metrics (TF-IDF: 93.41, Embed: 88.77, Jaccard: 77.05), whereas larger batches (10 and 20) preserve faithfulness more reliably, with batch size 20 reaching the highest values (97.77, 95.95, 86.19). This suggests that overly localized modifications drift from the original semantics, while larger batches constrain edits through signals aggregated across pages but dilute attention to page-specific characteristics. Overall, a batch size of 5 strikes the best trade-off, achieving the highest citation rate while maintaining competitive contribution (> 23) and faithfulness (> 83) across all sub-metrics.

F Citation Failure Taxonomy

This section details our taxonomy of citation failures, categorized into four primary modes: Technical Integrity, Semantic Alignment, Content Quality, and Systemic Exclusion. Based on an analysis of

Table 14: Citation rates and false-claim propagation before and after AgentGEO optimization on 130 evaluation cases.

Metric	Original	AgentGEO
Citation Rate	96/130 (73.85%)	111/130 (85.38%)
False Claim Among Cited Answers	6/96 (6.25%)	6/111 (5.41%)
False Claim Given Fact Addressed	6/31 (19.35%)	6/36 (16.67%)

Table 15: Comparison of optimization performance across different memory settings and batch configurations.

	Citation	Contribution			Faithfulness		
Config	**CR**	**Word**	**Pos**	**Wordpos**	**TF-IDF**	**Embed**	**Jaccard**
Vanilla	56.58	16.28	16.28	16.39	-	-	-
w/o Memory	75.94	22.21	22.28	22.09	96.89	94.39	84.51
w/ Memory	**79.32**	**22.69**	**22.66**	**22.53**	**96.95**	**95.45**	**84.86**
Batch 1	74.53	**25.56**	**25.57**	**25.52**	93.41	88.77	77.05
Batch 5	**80.86**	23.39	23.40	23.23	96.87	94.14	83.35
Batch 10	75.84	22.69	22.87	22.73	96.57	94.59	83.76
Batch 20	75.38	22.58	22.82	22.64	**97.77**	**95.95**	**86.19**

949 cases, the following subsections provide formal definitions for each mode, complemented by qualitative analyses of representative samples.

F.1 Technical Integrity

The page cannot be properly ingested by the retrieval pipeline. The failure occurs during fetching, rendering, or parsing, before semantic relevance or content quality can be evaluated.

F.1.1 JS/Dynamic Failure

The required content is absent from the extracted page because it depends on client-side rendering or SPA execution.

Query: *Where can I order a birthday cake online?*

Failure Analysis:
"Sorry, this webpage requires JavaScript to function correctly. Please enable JavaScript in your browser and reload the page."

F.1.2 Access Blocking

The crawler cannot access the page because of a firewall, 403 response, login wall, connection failure, or similar access restriction.

Query: *the federal communications commission (fcc) controls and regulates*

Failure Analysis:
"Due to aggressive automated scraping of FederalRegister.gov and eCFR.gov, programmatic access to these sites is limited to access to our extensive developer APIs. If you are human user receiving this message, we can add your IP address to a set of IPs that can access FederalRegister.gov
textbackslash
textbackslash& eCFR.gov; complete the CAPTCHA (bot test) below and click 'Request Access'."

F.1.3 Unparseable Content

The extracted content is empty, corrupted, structurally broken, or unreadable due to parsing failure.

Query: *Animal Castration – Why doesn't it seem to hurt very much?*

Failure Analysis:
Document B's hidden context is a stream of corrupted, binary-like characters and gibberish (e.g., `"jcs9m??j+?"??3?I?S?'???@\\%]r??3Pa ???F???Ã?99Z?(U?-P?=a‘Q?d?|Pb?y?2?? 0e??8M?uTDn?s\\$e?? ..."`), indicating a parsing failure or corrupted extraction that rendered the text unreadable and unusable. The system explicitly states it "could not generate refined text" due to unreadability.

F.1.4 Low Signal-to-Noise

The extracted page is dominated by ads, navigation, boilerplate, or metadata rather than substantive content.

Query: *who plays jekyll in once upon a time*

Failure Analysis:
Document B consists solely of metadata headings such as "Release Dates|Official Sites|Company Credits|Filming
textbackslash
textbackslash& Production|Technical SpecsRelated lists from IMDb users" without any substantive paragraphs or relevant content about the character or actor playing Jekyll. This indicates the content is overwhelmed by boilerplate or navigation elements rather than meaningful text.

F.2 Semantic Alignment

The document's content reaches the model but does not match what the query requires. It may be topically related while lacking the specific fact, entity, or temporal scope needed for the answer.

F.2.1 Intent Divergence

The document addresses a different user goal than the query, such as informational content for a transactional request.

Query: *when does the turn of the screw take place*

Failure Analysis:
Document B focuses almost exclusively on the 1974 film adaptation of *The Turn of the Screw*, detailing cast, production, broadcast, and reception of the movie. It does not provide any information about the original story's setting or timeline. For example, it states: > "The Turn of the Screw is a 1974 American made-for-television horror film directed by Dan Curtis based on the 1898 novella of the same name by Henry James." and the plot summary is limited to a general description of the film's story without specifying when the narrative takes place.

F.2.2 Contextual Gap

The document covers the general topic but omits the specific entity, concept, or terminology required by the query.

Query: *mention the names of any 3 famous folklore sports in karnataka state*

Failure Analysis:
Document B extensively discusses popular modern and professional sports in Karnataka such as cricket, football, badminton, hockey, kabaddi, tennis, and surfing, but it does not explicitly mention or name any traditional or folklore sports native to Karnataka. The only indirect references to traditional sports appear buried in the hidden context (e.g., "Traditional sports like Kambala—buffalo races contested in flooded paddy fields—and Korikatta (cockfighting) are very popular in the city"), but these mentions are minimal, lack detail, and are not clearly highlighted as famous folklore sports statewide. The main visible text focuses on mainstream, contemporary sports and notable athletes, failing to address the user's specific request for folklore sports names.

F.2.3 Outdated Information

The document is temporally mismatched with the query or relies on obsolete facts.

Query: *the most recent technological change to the u.s. economy was*

Failure Analysis:
Document B is a publication from 1987 titled *Technology and Employment: Innovation and Growth in the U.S. Economy*. It discusses technological change and employment effects from a historical perspective relevant to that time, focusing on issues like worker displacement, competitiveness, and policy recommendations from over three decades ago. There is no mention of recent technological changes such as AI, robotics, or the Fourth Industrial Revolution.

F.2.4 Localization Mismatch

The document targets the wrong region, jurisdiction, or language for the query.

Query: *Book a spa appointment for a facial treatment*

Failure Analysis:
Document B focuses exclusively on facial spas and treatments located in New York City, e.g., "The Best Facials in NYC," "Rescue Spa," "New York's Best Facialist Danuta Mieloch," and multiple NYC-based media endorsements.

F.3 Content Quality

The webpage addresses the right topic, but presents the required information poorly. The answer may be buried in irrelevant material, scattered across readable fragments, diluted by verbosity, or encoded in dense tables.

F.3.1 Information Scarcity

The text is too short, generic, or promotional to provide citeable facts for the query.

Query: *when is an articulated lorry most likely to jackknife*

Failure Analysis:
"All questions in this Heavy vehicle section (Loading) are listed below. If you click on a question it will show you the possible answers that you might be asked in a theory test." — This text is extremely short, generic, and does not provide any substantive information about when an articulated lorry is most likely to jackknife. Document A directly explains the conditions leading to jack-knifing, including the effect of load and braking on trailer behavior, providing a clear, relevant answer. Document B, by contrast, only references a list of questions without delivering any actual explanatory content, making it unhelpful for answering the user's query.

F.3.2 Content Fragmentation

The extracted text is readable, but the relevant content appears as disconnected snippets with poor narrative flow, making it difficult to synthesize a clear answer.

Query: *What is the role of gut bacteria in human health?*

Failure Analysis:
Document B contains numerous incomplete sentences, abrupt topic shifts, and fragmented ideas, e.g., T̈he microbiota consists of aThe microbiomeThis article explores the differences between the microbiota and microbiome.änd Ä person's gut microbiota develops from the first microbial exposure and typically reaches a full composition atBehaviorA person's gut microbiota and brain constantly communicate with each other. The gut-brain axisHowever, researchers have alsoDiseaseResearch suggests that bacterial populations in the gastrointestinal system play a role inThe status of the gut microbiotaDisturbing the microbiota with antibiotics can also lead to disease,ẅhich shows poor narrative flow and broken syntax.

F.3.3 Excessive Verbosity

The document contains relevant facts but dilutes them with rambling, repetitive, or overly legalistic prose.

Query: *when did the movie breakfast club come out*

Failure Analysis:
The losing document contains a long, rambling critique filled with repetitive, overly detailed, and meandering commentary on the film's pacing, tone, and acting, e.g., M̈ore detrimental to momentum is, of course, an unevenness in pacing... thoughtfulness often runs out of material... draggy written storytelling... unsubtly cheesy spots in dialogue and moments of melodrama... lapses in subtlety... limitations in acting material...T̈his verbosity dilutes the focus on the key factual information the user requested.

F.3.4 Unstructured Layout (Author Error)

The relevant content is organized in a dense or poorly signposted layout, such as a wall of text or raw table. It lacks clear summaries, headings, or direct answer statements.

Query: *what position did doug peterson play in the nfl*

Failure Analysis:
Document B consists almost entirely of dense, complex statistical tables with minimal narrative or explanatory text. The data is presented in raw tabular form with many abbreviations and no clear, readable summary or direct statement of Doug Pederson's position. For example, the document opens with a large passing statistics table and scattered fragmented stats without contextual sentences: > "Passing- > | Year | Age | Tm | Pos | No. | G | GS | QBrec | Cmp | Att | Cmp
textbackslash
textbackslash> ... > "Career| 100 | 17 | 3-14-0| 286 | 522 | 54.8 | 2762| 12 | 2.3 | 19 | 3.6 | 126 | 36.5 | 84 | 5.3 | 4.1 | 9.7 | 27.6| 62.3| 40 | 244 | 7.1 | 4.48| 3.39 | 1 | 1 | 3 |"
There is no straightforward sentence explicitly stating "Doug Pederson played as a quarterback in the NFL," which is the direct answer to the user query.

F.4 Systemic Exclusion

The target document is extractable, relevant, and usable, yet faces structural disadvantages beyond its content. This reflects system-side constraints, including authority-based redundancy and context-window truncation, rather than document-side defects.

F.4.1 Competitive Redundancy

Multiple documents provide similar information, but the generative engine cites a more authoritative competing source.

Query: *What is the theory of gravity wave detection?*

Failure Analysis:
Document B contains an extensive, detailed, and comprehensive exposition on gravitational-wave detection, covering resonant mass antennas, laser interferometers, historical context, technical challenges, and future detectors. The "Loser - As seen by System" snippet is identical in content to Document A, and the "Hidden Context" reveals even more in-depth information, including technical details, historical experiments, and advanced detector designs. This indicates Document B is valid, relevant, and rich in content.

F.4.2 Window Truncation

The answer exists in the document but falls outside the portion retained by the generative engine.

Query: *When is the snowboard big air competition?*

Failure Analysis:
The visible portion of Document B only contains a brief mention of an athlete's background: "Grace Henderson has been on the pro slopestyle rookie team since 2017 and has risen through the ranks, now competing on the pro team." The actual event schedule with dates and details about the snowboard big air competition is only found in the hidden context: "Visa Big Air Presented By Toyota COPPER MOUNTAIN, COLORADO Women and Men Big Air Event Schedule Copper Mountain, CO | December 13-16, 2023 Wednesday, December 13 | Snowboard Big Air Qualifiers Friday, December 15 | Snowboard Big Air Finals"

G Utility Tools for Textual Transformation

G.1 Optimization Tools Categories and Definitions

Following the diagnostic failure reasons identified in Section 3.2, our system employs a suite of **7 specialized optimization tools**. The specific prompt used for tool selection is presented in Table H.5.

Algorithm 1 Workflow of `AgentGEO`

Input: Target webpage d, Training query set Q_{tr}, Batch size B, Generative engine $\mathcal{G}$, Tool library Ω, Max iterations T_{max}
Output: Optimized webpage d^*

Initialize $d^* \leftarrow d$
Partition Q_{tr} into batches $\mathcal{B} = \{B_1, B_2, \ldots, B_K\}$ of size B
for each batch $B_k \in \mathcal{B}$ **do**
  Initialize Candidate Edits Set for current batch: $\mathcal{E}_{batch} \leftarrow \emptyset$
  ▷ ***Stage 1: Diagnose-then-Repair Loop***
  for each training query $q \in B_k$ **do**
    if $\mathcal{V}(q, d^*, \mathcal{G}) == 1$ **then** ▷ *Already cited, no optimization needed*
      continue
    Initialize surrogate webpage $d^{(t)} \leftarrow d^*$ ▷ *Isolated copy*
    Initialize query-specific memory $\mathcal{M}_q \leftarrow \emptyset$
    for $t = 1$ **to** T_{max} **do**
      if not cited ($\mathcal{V}(q, d^{(t)}, \mathcal{G}) == 0$) **then**
        Compare $d^{(t)}$ against one cited competitor to diagnose $v_q^{(t)}$
        Select repair tool $\omega^{(t)} \in \Omega$ using policy $\pi(v_q^{(t)}, \mathcal{M}_q, \Omega)$
        Produce modified webpage $d^{(t)}$ by applying tool $\omega^{(t)}$
        Update memory $\mathcal{M}_q \leftarrow \mathcal{M}_q \cup \{(v_q^{(t)}, \omega^{(t)})\}$
      else
        Extract edit suggestion s_q (modification, chunk index, HTML fragment)
        $\mathcal{E}_{batch} \leftarrow \mathcal{E}_{batch} \cup \{s_q\}$
        break
  ▷ ***Stage 2: Aggregate Across Queries***
  Partition d^* into structure-aware chunks using HTML tags
  for each targeted chunk index i **do**
    Group candidate edits $\mathcal{E}_i \subseteq \mathcal{E}_{batch}$ mapped to chunk i
    if $\mathcal{E}_i \neq \emptyset$ **then**
      Resolve conflicts to select highest-ranked edit $edit^*$
      Apply $edit^*$ to chunk i of d^* ▷ *Updates d^* for the next batch*
return d^*

These tools are designed to address specific document deficiencies through four functional categories: Information Augmentation, Structural Enhancement, Content Positioning, and Persuasive Refinement.

Category I: Information Augmentation Tools. This category addresses deficiencies in information density and completeness by injecting or restructuring missing content.

- `entity_injection`: This tool surgically inserts specific missing facts or entities into existing content. It performs "micro-insertion" at the sentence level using semantic analysis to identify optimal injection points, wrapping new entities in `<strong>` tags for semantic highlighting.
- `data_serialization`: Designed to improve parseability, this tool converts narrative-style data descriptions into structured HTML tables. It detects repeating patterns in prose and generates standard `<table>` elements with headers, while preserving non-tabular commentary.

Category II: Structural Enhancement Tools. These tools improve document organization to enhance both human readability and machine parseability without altering the core narrative.

- `structure_optimization`: This tool transforms unstructured "walls-of-text" by inserting hierarchical headers (e.g., `<h3>`), converting list-like sentences into `<ul>` or `<ol>` elements, and adding emphasis tags to key entities.
- `noise_isolation`: To assist parsers in distinguishing core content from boilerplate, this tool wraps non-semantic elements (navigation, ads, footers) in containers like `<aside>` or `<footer class="geo-noise">`, while designating high-value content with `<article>` tags.

Category III: Content Positioning Tools. The objective here is to restructure the document flow to surface relevant information, addressing answer positioning failures.

- `bluf_optimization`: Implementing the "Bottom Line Up Front" principle, this tool extracts key takeaways and generates a 1-2 sentence factual summary wrapped in a `geo-summary-box` at the top of the document.
- `content_relocation`: This tool surfaces hidden or truncated content by creating style-matched summary sections (e.g., "TL;DR" or "Key Findings") at the document header, synthesizing deep content into bullet points.
- `intent_realignment`: This operator rewrites the opening paragraph to directly address the user's query intent. It analyzes the query type and modifies the first sentence to provide a direct answer, moving unrelated details to the end rather than deleting them.

G.2 Hierarchical Policy Implementation Details

This section details the implementation of the selection policy $\pi(v_q^{(t)}, \mathcal{M}_q, \Omega)$ introduced in the main text. The policy is realized through a deterministic engine that maps the diagnostic feedback v_q to an operator $\omega \in \Omega$, subject to constraints imposed by the memory $\mathcal{M}_q$.

G.2.1 Diagnostic-to-Operator Mapping

The policy π prioritizes tool selection based on the severity and type of the vulnerability. We categorize operators into **Mandatory Corrections** (for technical/structural failures) and **Heuristic Refinements** (for semantic/qualitative failures). The mapping logic is defined as follows:

- **Structural Remediation (Mandatory):** Diagnoses indicating broken document integrity trigger forced operations. specifically, `CONTENT_TRUNCATED` triggers `content_relocation` to synthesize omitted text, while `WEB_NOISE` triggers `noise_isolation` to restore the signal-to-noise ratio.
- **Information Augmentation:** When v_q indicates `MISSING_INFO`, the policy prioritizes factual injection via `entity_injection` (for discrete entities) or `data_serialization` (for unstructured data).
- **Positioning & Intent:** Failures related to answer visibility (e.g., `BURIED_ANSWER`) or intent misalignment (`SEMANTIC_IRRELEVANCE`) invoke `bluf_optimization` or `intent_realignment`, relocating critical information to the document prefix (BLUF).
- **Persuasive Refinement:** For high-level failures involving `TRUST_CREDIBILITY` or `OUTDATED_CONTENT`, the system applies soft optimization strategies such as `persuasive_rewriting` or `historical_redteam`.

G.2.2 Memory-Driven Constraints

To ensure convergence and prevent cycles, the policy utilizes the trajectory memory $\mathcal{M}_q = \{(v_q^{(k)}, \omega^{(k)})\}_{k=1}^{t-1}$ to enforce the following constraints:

1. **Idempotency Guard:** An operator ω is masked from the selection set Ω if $(\cdot, \omega) \in \mathcal{M}_q$ resulted in the same vulnerability v_q in a previous step.
2. **Budget Threshold:** To avoid semantic drift from over-optimization, any specific operator is restricted to a maximum of three invocations per query trajectory.
3. **Stability Pruning:** If an operator results in consecutive citation failures across two distinct optimization cycles, it is globally precluded from the current selection path.

G.2.3 Escalation Protocols

The policy π is adaptive; if the initial operator selection fails to resolve the vulnerability (i.e., t increases), the system escalates the strategy:

1. **Structure $\rightarrow$ Visibility:** If standard re-organization fails to address a `BURIED_ANSWER`, the policy defaults to `bluf_optimization` to force-surface the key takeaway in a summary box.
2. **Density Adaptation:** Upon repeated truncation alerts, the policy triggers `noise_isolation` with aggressive parameters to prune non-semantic boilerplate, maximizing the token budget for relevant content.

H Prompt Templates

This appendix provides a comprehensive repository of the prompt templates utilized within our framework. These prompts facilitate the interaction between the optimization controller and the large language model (LLM), governing the entire pipeline from diagnosis to execution and final response synthesis.

H.1 Failure Taxonomy

For each user query, we construct comparative document pairs consisting of a winning document and a rejected losing document to diagnose why the latter was disfavored by the generative retrieval system. An explainer module first generates a fine-grained diagnostic analysis of the comparative shortcomings, from which an extractor distills atomic, generalizable failure principles that objectively characterize the rejection determinants. Once accumulated across all pairs, these raw rules undergo operational-equivalence-based consolidation via a merger to derive canonical failure categories, followed by a filtering stage that strips query-specific dependencies to yield the final, context-independent taxonomy.

Prompt: Explain

[Task] You are an expert AI analyst. Your task is to analyze two documents that were retrieved by a search engine (and by extension, LLMs acting as retrieval engines) to answer a user's query. These documents were crawled from the web.
One document ("the winning document") was heavily used by the generative engine to generate its final answer, indicating a higher relevance or quality. The other document was used less.
Please provide a detailed explanation for why the loser document was less favored by the RAG system compared to the winning document.
Consider factors such as:
1. Web Crawling & Parsing Integrity:
- Render Failure: Is the content missing because it required JavaScript (e.g., "Loading...", empty body)?
- Boilerplate Noise: Is the text dominated by non-content elements (nav bars, footers, "cookie consent", "subscribe" popups, JSON code) rather than the article body?
- Formatting Loss: Has a data table or list been flattened into an unreadable paragraph?
2. Semantic Relevance & Alignment:
- Query Intent: Does the document address the core intent (informational, navigational, transactional) or just match keywords loosely?
- Entity Matching: Does it contain the specific entities (names, dates, prices, technical terms) requested, or is it vague?
- Directness: Does the answer appear in the first few paragraphs (top-heavy), or is it buried deep after irrelevant intros?
3. Content Quality & Structure:
- Information Density: Is the text "fluffy" (marketing jargon, filler) with low signal-to-noise ratio, compared to the winner's fact-dense content?
- Structure: Does it use headers, lists, and clear segmentation that helps the RAG system chunk the text effectively?
- Completeness:*
{rules_block} [User Query] {query}
[Document A] {winner_content}
[Document B] {loser_content}
[Winning Document]: {winner_content}
[Your Explanation] Provide your analysis below, explaining the strengths of the winning document and the weaknesses of the other in relation to the user's query. Focus primarily on the shortcomings and deficiencies of Document B, specifically distinguishing whether it was rejected due to lack of semantic relevance or due to poor data quality from the crawling process (e.g., the text is just a "cookie consent" banner, navigation menu, or empty page).

Prompt: Extration

[Instruction] Based on the following explanation about why <Loser Document> was not cited, extract a set of general, reusable reasons that define why a source document is rejected by a RAG system. These reasons should be objective and deterministic principles explaining the document's failure.
[Guidelines for Extraction]
1. Generalize: Convert specific instances into broad principles (e.g., "Missing iPhone price" -> "Lacks specific attribute data").
2. Contextualize: You may include a specific example in parentheses to make the rule clear (e.g., "...lacks specific data (e.g., mentions the car's color but misses the requested '0-60mph' time)").
3. Distinguish: Clearly separate **Technical Failures** (crawling noise, empty pages) from**Semantic Failures** (irrelevant content, missing logic).
4. Format: Return the list STRICTLY as a JSON array of strings. Do not use Markdown code blocks.
[Examples of Valid Reasons]
["The document content consists primarily of web noise (e.g., cookie banners, navigation menus, login prompts) rather than semantic content.",
"The document is technically unreadable due to severe parsing artifacts (e.g., raw JSON blobs, CSS classes, or JavaScript code mixed with text).",
"The document addresses the correct entity but focuses on the wrong attribute (e.g., discussing the 'history' of the company when the query asks for 'current stock price').",
"The document shares a similar theme with the query but lacks the specific granular data or metrics required (e.g., provides qualitative descriptions instead of the requested numbers).",
"The document information is outdated or temporally mismatched (e.g., discussing '2020 tax rules' when the query implies current '2024 regulations').",
"The user query involves multiple distinct entities (e.g., 'Entity A' and 'Entity B'), but the document addresses only one of them, failing to provide the necessary context for both.",
"The user query requires a relationship or comparison between entities, but the document discusses them in isolation without establishing the link or contrast.",
"The document is topically irrelevant to the user query."
]
[Explanation] explanation

Prompt: Extraction

[System Task] You are an expert Logic Analyst for an SEO Pipeline.
Your objective is to consolidate a list of raw rejection reasons into a concise set of**Canonical Reasons** to inform the development of future optimization tools.
[Merge vs. Segregate Logic]
Determine whether to merge reasons by asking: **"Would the exact same coding logic or prompt operation fix both issues?"**
1.**MERGE: Operational Equivalence**
- *Example:* "Missing price" AND "Missing date".
- *Logic:* Both imply an **"Insert specific data"** operation. Even though the data differs, the mechanism (Insertion) is the same. Merge them.
- *Example:* "Cookie banner noise" AND "Navigation menu noise".
- *Logic:* Both imply a **"Delete/Clean pattern"** operation. Merge them.
2. **SEGREGATE: Operational Distinctness**
- *Example:* "Content is too short" vs. "Content is off-topic".
- *Logic:* - "Too short" requires **Expansion** (Generating new context).
- "Off-topic" requires **Replacement** (Deleting and writing new topic).
- These are different operations. Keep them separate so we can decide later if we need one tool or two.
- *Example:* "Structure is messy (prose)" vs. "Table is broken (data)".

- *Logic:* Prose needs **Segmentation** . Tables need **Schema Reconstruction** . Different mechanisms. Keep separate.
[Instruction on Output Format]
Return the merged list as a single, valid JSON array of strings. Do not use j̈sonör add explanations.
[Original Rules] rules
[Merged Rules JSON]

Prompt: Filter

[Persona] You are a technical writer specializing in creating context-independent documentation.
[Task] Analyze the following rule. Your goal is to remove any part of the rule that makes it dependent on a specific user "query", "question", or "input". The rewritten rule should state a general principle.
- If the rule contains a general principle AND a reference to a query, remove only the query reference. - If the entire rule is ONLY about how to handle a query (e.g., "The document should directly answer the query."), the principle is not general. In this case, you should return an empty string.
[Polarity] Keep the original polarity. If the rule is written as an avoidance/negative statement (e.g., "do not", "should not", "avoid", "lacks"), do NOT flip it into a positive requirement. Preserve the “avoid/should not/lacks” wording when possible.
[Examples] - Input Rule: "The document should provide specific facts and data relevant to the user’s query." - Output JSON: "modified rule": "The document should provide specific facts and data."
- Input Rule: "The source must be recent and directly answer the question." - Output JSON: "modified rule": "The source must be recent."
- Input Rule: "The text must be authoritative." - Output JSON: "modified rule": "The text must be authoritative."
- Input Rule: "Directly answer the user’s question." - Output JSON: "modified rule": ""
[Instruction on Output Format] Return a single, valid JSON object with one key: "modified rule". The value should be the modified string.
[Input Rule] "rule"
[Output JSON]

H.2 Query Generation

This section details the implementation of the 5 specialized prompt templates within the query generator module. Each template is meticulously engineered to simulate authentic user search behaviors—spanning profile extraction, intent classification, and domain relevance filtering—while strictly enforcing persona-specific vocabulary and semantic consistency during the query generation pipeline.

Prompt: Keyword and Persona Extraction

You are a Senior Semantic SEO Specialist and User Intent Analyst. Your goal is to reverse-engineer the search intent behind a specific webpage title to uncover high-value keyword clusters and precise audience segments.

TASK:
Analyze the title to identify its core topics and assess the types of users who are likely to be interested in those topics. Crucially, you must expand these topics to include:

ANALYSIS REQUIREMENTS:
1. **Topic/Keyword Extraction:**
    - **Core Keywords:** The primary search terms.

- **LSI/Synonyms:** Synonyms and semantically related terms (Latent Semantic Indexing) that add context.
- **Keyword Phrases:** A specific, multi-word search query that reflects clear user intent.
- **Ambiguity:** If a keyword carries multiple meanings (polysemy), appropriately expand it by adding necessary modifiers to strictly define its specific meaning based on the web content. (e.g., clarify "Jaguar" to "Jaguar Car" or "Jaguar Animal").

2. **Persona Profiling:**
   - Identify distinct user personas
   - For each persona, identify:
     - **Role/Demographic:** Specific groups (e.g., Infants, Children, Teens, Adults, Seniors, Pregnant women).
     - **Description:** A concise overview of the persona's circumstances, offering general context without referencing the keyword specifically.

OUTPUT FORMAT:
Please present your analysis in the following JSON format:

```
{{
    "keyword_cluster": {{
    "core": ["keyword1", "keyword2"],
    "lsi_synonyms": ["synonym1", "synonym2"],
    "keyphrases": ["phrase1", "phrase2"]
    }},
    "target_personas": [
    {{
        "name": "Persona Name",
        "description": "Persona Description"
    }}
    ]
}}
```

Prompt: Intent Classification

You are a Senior SEO Specialist and Google Search Quality Rater. Your task is to analyze a website's **Title** and **Content Summary** to determine which User Search Intents this page satisfies.

THE 4 SEARCH INTENTS (DEFINITIONS):
You must select from the following four categories based on the user's goal:

1. **Informational (KNOW):** The user wants to learn something, find an answer, or solve a problem. (e.g., Guides, Tutorials, Definitions).
2. **Commercial Investigation (COMPARE):** The user is in the consideration phase, researching products/services but not ready to pay yet. (e.g., Reviews, Comparisons, "Best of" lists).
3. **Transactional (DO):** The user is ready to take a specific action, usually involving a conversion. (e.g., Buy, Subscribe, Download, Register).
4. **Navigational (GO):** The user is looking for a specific website or page. (e.g., Login pages, Contact pages, About Us).

TASK INSTRUCTIONS:

1. Analyze the provided **Title** and **Summary**.
2. Evaluate the page against **ALL 4 intents**.
3. **Filter Logic (Crucial):** Only assign an intent if the content provides a **substantial and satisfying solution** for that intent.
   - *Example:* A blog post with a small "buy now" link in the footer is **NOT** Transactional. It is Informational.

- *Example:* A product page with a short description is primarily Transactional, but NOT Informational (since it lacks depth).

4. **Output:** Return the applicable intents (can be multiple) with a confidence score and reasoning.

OUTPUT FORMAT:
List the suitable intents and provide a short reasoning based on why they were NOT excluded.

Output Format:
- **Intents:** [List of intents chosen from Informational, Commercial Investigation, Transactional, Navigational]
- **Reasoning:** [Brief reasoning explaining the primary goal of the content and any secondary goals based on specific keywords or content features.]

Prompt: Query Generation

You are a Realistic Search Engine User Simulator. Your goal is to mimic the exact search behavior, vocabulary, and intent of a specific user segment.

INPUT CONTEXT:
- **Focus Keyword:** `{keyword}`
- **User Persona Profile:**
  - **Role/Demographics:** `{persona_name}` (e.g., Senior Citizen, Python Developer, Busy Mom)
  - **User Description:** `{persona_description}` (Current situation, experience level, environment)

TASK:
Generate exactly 5 search queries related to `"{keyword}"` for EACH of the following applicable search intents.

APPLICABLE SEARCH INTENT CATEGORIES & GUIDELINES:
`{intent_section}`

OUTPUT FORMAT:
Return a JSON object with arrays for each intent type.
For intent types NOT listed above, return an empty array `[]`.

```
{{
    "navigational": ["query1", ...] or [],
    "informational": ["query1", ...] or [],
    "commercial": ["query1", ...] or [],
    "transactional": ["query1", ...] or []
}}
```

Prompt: Deduplication

You are an expert Semantic Query Optimizer and Data Cleaner. Your task is to analyze a list of user search queries and remove duplicates based on **User Intent** and **Semantic Equivalence**.

 # Core Objective
Reduce redundancy in the provided list while preserving every distinct user intent, specific entity (products/dates), and action type.

 # Deduplication Logic (Step-by-Step)
1. **Exact & Morphological Duplicates:**
   - Remove queries that are identical strings (case-insensitive).
   - Remove queries that are merely reordered variations of keywords without changing the meaning (e.g., "interior 2021 g-wagon" vs. "2021 g-wagon interior").

2. **Semantic Equivalence (Synonyms):**
   - Merge queries where keywords are swappable synonyms in the specific context.
   - *Example:* "Luxury cabin" ≈ "Premium interior".
   - *Example:* "Specs" ≈ "Specifications" ≈ "Features".
   - *Example:* "Buy" ≈ "Purchase".
3. **Representative Selection (The "Canonical" Query):**
   - When a group of similar queries is identified, select **one** representative query to keep.
   - **Priority for selection:**
     (a) The most grammatically natural and fluent option.
     (b) The most specific/descriptive option (provided it covers the intent of the others).
     (c) The option that uses standard industry terminology.

 # Strict Exclusion Rules (Do NOT Merge)
1. **Distinct Entities:** Do not merge queries referring to different models, versions, or years (e.g., "2021 model" is NOT the same as "2023 model"; "G-Class" is NOT the same as "GLE").
2. **Distinct Intents:** Do not merge different stages of the user journey.
   - *Transactional:* "Buy..."
   - *Informational:* "Reviews of...", "Specs of..."
   - *Navigational:* "Official site..."
   - *Comparison:* "Model A vs Model B"

 # Output Format
Return only the final deduplicated list of strings in valid JSON format: `["query1", "query2", ...]`

Prompt: Domain Filtering

You are an SEO Relevance Expert and Topic Cluster Specialist. Your goal is to filter a list of search queries based on their potential alignment with a specific document.

THE CORE OBJECTIVE:
You must act as a **Broad Filter**. Your job is NOT to find only the "perfect" matches. Your job is to **remove noise**.
- **Default Action:** KEEP the query.
- **Exception:** REMOVE only if the query is definitely irrelevant.

1. RETENTION RULES (When to KEEP):
- **Same Domain:** Keep all queries related to the document's general industry or topic (e.g., if Doc is about "SEO", keep "digital marketing" queries).
- **Low Relevance / Tangential:** Keep queries that are loosely related. If we could add a paragraph to the document to answer this query, KEEP IT.
- **Broad/Vague:** Keep broad queries (e.g., "what is marketing") if the document covers a specific aspect of that topic.
- **Competitor Comparisons:** Keep queries comparing the subject to alternatives.

2. EXCLUSION RULES (When to REMOVE):
Remove a query **strictly** if it meets one of these conditions:

• **Semantic Mismatch (Polysemy):** The query uses a word that appears in the document but refers to a completely different concept/industry.
- *Example (Syndication):* Document is about "Media Content Syndication".
- **KEEP:** "content distribution platforms" (Same domain).
- **REMOVE:** "bank loan syndication process" (Finance domain - completely different audience).
- *Example (Python):* Document is about "Python Programming".
- **REMOVE:** "types of python snakes in florida" (Zoology domain).

• **Wrong Industry/Intent:** The user searching for this has **zero** overlap with the target audience of the document.

- *Example:* Doc is about "Enterprise CRM Software"; Query is about "Free Minecraft skins" → REMOVE.

3. DECISION HEURISTIC:
Ask yourself: **"Could the person searching for [Query] potentially find value in [Document Content]?"**

- If YES (even "1%" chance): **KEEP**.
- If NO (impossible): **REMOVE**.

OUTPUT FORMAT:
The output queries must be equal to the input! Return the result in JSON format with two distinct lists:

```
{
  "relevant_queries": ["list", "of", "queries", "to", "keep"],
  "filtered_queries": ["list", "of", "removed", "queries"]
}
```

H.3 Search Failure Analysis

This section presents the prompt design for search failure analysis, a critical diagnostic step in the optimization pipeline.

Prompt: Search Failure Analysis

You are a Search Failure Analyst with expertise in document quality assessment. Your task is to determine the **ROOT CAUSE** of why the Target document was **NOT** cited while the Competitor was.

Query: `"{query}"`

=== COMPETITOR DOCUMENT (Winner - Was Cited) ===
`{competitor}`
=== END COMPETITOR ===

=== TARGET DOCUMENT (Loser - Not Cited) ===
`{target}`
=== END TARGET ===

`{failure_taxonomy}`

ANALYSIS INSTRUCTIONS:
1. **Compare** the two documents carefully against the query
2. **Identify** the PRIMARY reason the Target failed
3. **root_cause MUST be EXACTLY one of**:
   `PARSING_FAILURE, CONTENT_TRUNCATED, DATA_INTEGRITY, WEB_NOISE, LOW_SIGNAL_RATIO, LOW_INFO_DENSITY, MISSING_INFO, STRUCTURAL_WEAKNESS, SEMANTIC_IRRELEVANCE, ATTRIBUTE_MISMATCH, BURIED_ANSWER, NON_FACTUAL_CONTENT, TRUST_CREDIBILITY, OUTDATED_CONTENT, UNKNOWN`
4. **Be Specific** in explanation - identify WHAT specifically is missing or wrong

SEVERITY GUIDE:
- `critical`: Document is fundamentally unusable (parsing failure, completely irrelevant)
- `high`: Major issue requiring significant changes (truncation, buried answers, missing key info)
- `medium`: Moderate issue requiring targeted fixes (structure, density)
- `low`: Minor issue requiring polish (formatting, minor noise)

CRITICAL: Output `root_cause` as **EXACT** category name (e.g., `"MISSING_INFO"`, not `"Missing Information"`)

{format_instructions}

H.4 Tools Selection Prompt

This section provides prompt template for the Strategic Tool Selection module, designed to map diagnosed failure modes to the most effective optimization utilities

Prompt: Tools Selection

You are the GEO Optimization Controller. Select the best tool based on the Diagnosis and Policy.
Inputs
- **Query**: {query}
- **Root Cause Diagnosis**: {diagnosis_cause} ({diagnosis_explanation})
- **History**: {history_context}

Policy {policy_guidelines}
Target Document (Indexed) {target_content}
Available Tools {tool_descriptions}
Task Select the tool to resolve the '{diagnosis_cause}'. Identify the specific [CHUNK_ID] that needs modification.
Constraints
- **Targeting:** You must include `"target_chunk_index": <int>` in your output.
- **Injection Rule:** Include `["target_content", "context_before", "context_after", "core_idea", "previous_modifications"]` in `tool_arguments` with empty string values (`""`). The system will populate these.

{format_instructions}

H.5 Tools Prompt

This section provides the implementation details for the **9 specialized optimization tools**. Each template is engineered to address specific taxonomic failures—spanning information density, structural clarity, and persuasiveness—while strictly enforcing context preservation and semantic integrity during document transformation.

Example: Intent Realignment Transformation

Query: "How to reduce carbon footprint?"

Original: In today's rapidly changing world, environmental concerns are paramount...
Optimized: **Reducing your carbon footprint** can be achieved through three primary strategies: transportation changes, energy efficiency, and consumption habits...

Module: History Section

PREVIOUS MODIFICATIONS (MUST PRESERVE):
{previous_modifications}

Prompt: BLUF Optimization Tool

You are a UX Writer for Information Retrieval. Surface the key takeaway immediately using the BLUF (Bottom Line Up Front) principle.
Key Takeaway to Surface: {key_takeaway}
{context_section}

Core Idea: {core_idea}
{history_section}
SPECIFIC INSTRUCTIONS:

1. Generate the HTML content for a summary box.
2. Summarize the "{key_takeaway}" in one direct, factual sentence using .
3. Do **NOT** output the existing content, **ONLY** the new summary content.

OUTPUT FORMAT:

```
[HTML Content for the BLUF Summary Div ONLY]

--MODIFICATION_SUMMARY--
- [Change 1]
```

Prompt: Content Relocation Tool

You are a Content Restructuring Expert. The document has **CRITICAL** information buried deep (truncated from view). Your job is to **SURFACE** this information by creating a prominent summary section at the TOP.

Context

- **User Query:** {query}
- **Core Idea:** {core_idea}
- **Problem:** Important information is truncated and invisible.

Hidden Content Summary(from truncated section)
{hidden_summary}
{context_section}
{history_section}
{preservation_rules}

STEP 1: Analyze Content Style
First, analyze the target content to determine its style:

- **Formal/Academic:** Research papers, official docs, business reports → Use "Summary" or "Key Findings"
- **Technical/Documentation:** API docs, tutorials, technical guides → Use "Quick Answer" or "Key Points"
- **Blog/Casual:** Personal blogs, forums, community posts → Use "TL;DR" or "The Short Version"
- **News/Editorial:** News articles, reviews, editorials → Use "Highlights" or "At a Glance"
- **E-commerce:** Product pages, listings → Use "Quick Facts" or "Overview"

STEP 2: Create Summary Section

1. **Choose the most appropriate header** based on the content style detected above.
2. **Synthesize** the hidden content summary into 2-4 bullet points that DIRECTLY answer the query.
3. **Use semantic HTML:** Wrap in , use <h3> for header, for points.
4. **Match the tone:** The summary should feel native to the page - formal content gets formal language, casual content gets casual language.
5. **Preserve ALL original content** below this new section - DO NOT delete anything.

STEP 2: Style Matching Rules

1. Be concise but include specific facts, numbers, entities from the hidden summary
2. Mirror the vocabulary and tone of the original content
3. The new section should blend seamlessly with the existing page

=== TARGET CONTENT (Current Visible) ===
{target_content}
=== END TARGET CONTENT ===

OUTPUT FORMAT:

```
[New Summary Section with style-appropriate header]
[Original Content Preserved Below]

--MODIFICATION_SUMMARY--
- Detected content style:  [style type]
- Added "[chosen header]" section at top with [specific facts
surfaced]
```

Prompt: Data Serialization Tool

You are a Data Structuring Agent. Your task is to convert narrative descriptions of data into structured HTML Tables.
{context_section}
Core Idea: {core_idea}
{history_section}
{preservation_rules}
SPECIFIC INSTRUCTIONS:

1. **Identify Patterns:** Identify repeating data patterns in the text (e.g., "Product A costs $10. Product B costs $20.").
2. **Create Table:** Create an HTML `<table>` with appropriate `<thead>` headers to represent this data.
3. **CRITICAL:** Do **NOT** delete the original narrative text completely if it contains nuance/sentiment that fits poorly in a table. Instead, keep the text as a summary *after* the table, or integrate the nuance into a "Notes" column.
4. **Completeness:** Ensure every number and entity from the text exists in the table.

=== TARGET CONTENT (HTML Fragment) ===
{target_content}
=== END TARGET CONTENT ===

OUTPUT FORMAT:

```
[HTML Content with Table...]

--MODIFICATION_SUMMARY--
- [Change 1]
```

Prompt: Entity Injection Tool

You are a Content Enricher and HTML Editor. Your task is to perform a "surgical injection" of missing information into the provided HTML fragment.
Missing Entities to Inject: {missing_entities}
{context_section}
Core Idea: {core_idea}
{history_section}
{preservation_rules}
SPECIFIC INSTRUCTIONS:

1. **Locate the Best Spot:** Analyze the HTML content to find the most semantically relevant place.
2. **Rewriting:** Rewrite the relevant sentence/paragraph to naturally include the new fact.
3. **Semantic Highlighting:** Wrap the injected entity in `<strong>` tags.
4. **Preserve Structure:** Do **NOT** remove existing links (`<a href...>`) or formatting unless necessary for the rewrite.
5. **Output:** Return the FULL modified HTML chunk.

=== TARGET CONTENT (HTML Fragment) ===
{target_content}
=== END TARGET CONTENT ===

OUTPUT FORMAT: First output the enhanced HTML content, then the summary.
[Enhanced HTML content...]

--MODIFICATION_SUMMARY--
- [Inserted '{missing_entities}']

Prompt: Intent Realignment Tool

You are an expert SEO Content Editor. Your goal is to make the content explicitly answer the User Query immediately, without losing original depth.
User Query: "{user_query}"
{context_section}

Core idea: {core_idea}
{history_section}

{preservation_rules}
SPECIFIC INSTRUCTIONS:

1. Analyze the intent of the query (Informational/Navigational/Commercial).
2. Rewrite the specific TARGET CONTENT to address this intent in the first sentence.
3. Remove "fluff" or vague introductions (e.g., "In today's world...").
4. Ensure keywords related to the query appear naturally.
5. **CRITICAL**: If the original content contains specific details (dates, names, specs) not related to the query, MOVE them to the end of the chunk, DO NOT DELETE THEM.

=== TARGET CONTENT (HTML Fragment) ===
{target_content}
=== END TARGET CONTENT ===

OUTPUT FORMAT: First output the enhanced HTML content, then the summary.
[Enhanced HTML content...]

--MODIFICATION_SUMMARY--
- [Change 1]

Prompt: Noise Isolator Tool

You are an HTML Semantic Cleaner. Your job is to tell search engine parsers what is NOISE and what is CONTENT.
{context_section}
Core idea: {core_idea}
{history_section}
{preservation_rules}
SPECIFIC INSTRUCTIONS:

1. Analyze the target content. Identify elements that are likely:
   - Navigation links
   - Cookie warnings
   - Sidebar ads / "Read More" links
   - Copyright footers
2. Wrap these elements in <aside>, <nav>, or <footer class="geo-noise"> tags.
3. Wrap the actual high-value semantic content in <article> or <section>.

4. **DO NOT DELETE** the noise (as it might be functionally needed for the webpage UI), just wrap it semantically so parsers can de-prioritize it.

=== TARGET CONTENT (HTML Fragment) ===
{target_content}
=== END TARGET CONTENT ===

OUTPUT FORMAT:

```
[Semantically Wrapped HTML...]

--MODIFICATION_SUMMARY--
- [Change 1]
```

Prompt: Static Renderer Simulator

You are a Server-Side Rendering (SSR) Simulator. The input content contains JavaScript code, JSON blobs, or "Loading..." placeholders that hide the actual content. Your task is to "render" this into static, readable HTML text based on what the JS code implies.
{context_section}
Core idea: {core_idea}
{history_section}
{preservation_rules}
SPECIFIC INSTRUCTIONS:

1. Look at the JS code or JSON data in the input.
2. Extract the actual human-readable content (text, prices, items) from the code logic.
3. Output the content as plain, static HTML tags (<p>, <ul>,).
4. **REMOVE** the <script> tags and raw JSON.
5. If the JS implies fetching external data that is NOT present in the chunk, place a placeholder: [Data to be fetched] but do not hallucinate data.

=== TARGET CONTENT (Raw/JS-heavy Chunk) ===
{target_content}
=== END TARGET CONTENT ===
OUTPUT FORMAT:

```
[Static HTML Content...]

--MODIFICATION_SUMMARY--
- [Change 1]
```

Prompt: Structure Optimization Tool

You are an HTML Semantic Architect. Transform the unstructured text into parser-friendly, structured HTML.
{context_section}

Core idea: {core_idea}
{history_section}

{preservation_rules}
SPECIFIC INSTRUCTIONS:

1. **Semantic Hierarchy**: Insert <h3> or <h4> tags (based on context depth) to break up long text blocks.
2. **List Conversion**: Detect sentences that list items (e.g., "We have A, B, and C") and convert them into or .
3. **Emphasis**: Wrap key entities (defined in the text) with or .

4. **NO TEXT CHANGES**: You are strictly forbidden from changing the wording, tone, or length of the text. ONLY add HTML tags.

=== TARGET CONTENT (HTML Fragment) ===
`{target_content}`
=== END TARGET CONTENT ===
OUTPUT FORMAT:
`[Structured HTML content...]`

`--MODIFICATION_SUMMARY--`
`- [Change 1]`

H.6 Citation and Answer Generation

To assess optimization efficacy across diverse search engine paradigms, we utilize distinct templates for citation-anchored generation:

- **In-Content Citation Generator:** The LLM receives all retrieved documents concurrently, generating the complete response with integrated citations in a single inference step.
- **Attr-first-then-generate Citation Generator:** A staged pipeline comprising: (1) *Content Selection*—extracting and highlighting core evidentiary sentences from each document; (2) *Attribution clustering*—sequencing selected evidence into thematic clusters aligned with the target response structure; and (3) *Structured Generation*—synthesizing content cluster-by-cluster to construct the final answer while maintaining explicit source attribution throughout.

These standardized templates ensure a consistent baseline for evaluating the performance gains achieved through our document optimization pipeline.

Prompt: In-Content Citation Generator

System Instruction:
Write an accurate and concise answer for the given user question, using *only* the provided summarized web search results. The answer should be correct, high-quality, and written by an expert using an **unbiased and journalistic tone**. The user's language of choice (e.g., English, Français, Español, Deutsch, or Japanese) should be used.
The answer logic must be rigorous and defensible. Every sentence in the answer should be ***immediately followed*** by an in-line citation to the search result(s) (e.g., `[index]`). The cited search result(s) should fully support *all* the information in the sentence.When citing several results, use the `[1][2][3]` format rather than `[1, 2, 3]`. You can use multiple search results to respond comprehensively while avoiding irrelevant search results.
Human: Question: `{query}`

Search Results: `{source}`

Prompt: Attr-first-then-generate Citation Generator

Content Selection:
In this task, you are presented with a question and several text chunks (sentences). The goal is to identify the minimal number of chunks that are relevant to answer the question.

- Return **ONLY** the indices of the selected chunks as a comma-separated list.
- **IMPORTANT:** The output must be strictly formatted as: `[chunk_index1, chunk_index2, ...]`.
- **IMPORTANT:** Chunk indices must follow the format `'doc_id-chunk_id'` (e.g., `0-0`, `0-1`, `1-0`).
- **IMPORTANT:** Select approximately **5–15 chunks** that are most relevant to the question.

Question: {query}
Chunk [0-0]: {content_of_chunk_0}
Chunk [0-1]: {content_of_chunk_1}
...
Chunk [doc_id-sent_id]: {content_of_chunk_n}

Answer (comma-separated chunk indices only):

Clustering:
In this task, you are presented with a question and several passages, where some parts are "highlighted" (namely, there are {<highlight_start>} and {<highlight_end>} tokens before and after each such span). Those spans are supposed to have the information needed to answer the question, and the goal is to fuse all those highlights into a single paragraph that answers the question.
As an intermediate step, you need to cluster highlights that can be merged into a sentence (namely, each cluster will be later merged into one sentence). Make sure the clusters are in the same order you would then write the corresponding output paragraph.

- **IMPORTANT:** make sure there are at least two clusters.
- **IMPORTANT:** The output must be of the format [{"cluster":[comma-delimited highlights indices]}].

Question: {query}
Content:
... text {<highlight_start>} selected span {<highlight_end>} text ...

Answer

Generation:
In this task, you are presented with a question and several passages, where some parts are "highlighted" (namely, there are {<highlight_start>} and {<highlight_end>} tokens before and after each such span). You are also presented with a prefix of a paragraph.
The goal is to generate a paragraph that answers the question, based on the passages. The given prefix is what you generated so far. Your job is to generate the **next sentence** in the paragraph, that covers all and only the "highlighted" spans.

- Make sure it connects well with the prefix.
- Ensure that it covers **all and only** the "highlighted" spans.
- Ensure that it properly builds towards a full and coherent response to the question.

Question: {query}
Content:
... text {<highlight_start>} selected span {<highlight_end>} text ...
Prefix: {prefix} *(Note: Empty if generating the first cluster)*

Answer:

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state three contributions: (i) a taxonomy of citation failure modes (Section 3, Figure 2), (ii) AgentGEO, a diagnostic agentic framework (Section 4), and (iii) the MIMIQ document-centric benchmark (Section 5). Empirical claims of 15-40% relative improvement and 5% content modification are supported by Tables 1, 2, and 3.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Limitations are discussed throughout the paper: Section 6.5 examines failure cases where content-side optimization cannot overcome source-level priors; Section 6.4 identifies topic categories where gains saturate

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: All experimental details required for reproduction are provided.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and data are released at `https://zhihuat.github.io/agentgeo/`

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experimental settings are detailed in Section 6.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Main results in Tables 1-3 are reported as point estimates without error bars due to the cost of multi-seed runs across the full benchmark (204 documents × 60 queries × multiple GE configurations).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute requirements are characterized in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Both positive and negative societal impacts are discussed in Appendix (Broader Impacts).

    Guidelines:

    - The answer [Yes] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: [N/A]

Guidelines:

- The answer [Yes] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The MIMIQ benchmark is a new asset introduced in this paper.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects. Annotation in Section 3.1 was performed by three of the authors (graduate students in NLP) for taxonomy validation, not by external crowd workers.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.