H-TSO: Hierarchical Black-Box Search for Minimal Context Attribution in Closed Language Models

Anonymous ACL submission

Abstract

Context attribution for closed LLMs requires identifying which parts of a long prompt support a generated answer when gradients and token log-probabilities are unavailable. We study this as a generation-only subset-search problem with binary sufficiency feedback. We propose H-TSO, a hierarchical black-box attribution method that combines role-conditioned priors, hierarchical Thompson sampling with greedy shrinking, and an explicit pair-interaction test for multi-hop evidence. On 300 stratified English LongBench-Cite examples, with `gpt-4o-mini` as both the target generator and search-time binary judge, and GPT-4o as the final citation-quality evaluator, H-TSO improves over Direct LLM selection (0.832 vs. 0.818 citation F1; paired bootstrap CI $[+0.004, +0.030]$) and comes within 2.8 F1 points of BM25 top-10 (0.860) while using roughly half as many selected sentences (5.3 vs. 9.8). Ablations show that hierarchical scoping and role-conditioned priors provide the largest gains, while pair-interaction testing helps multi-evidence examples. We analyze judge sensitivity with a 50-example human audit and identify remaining limitations in target-model coverage and triplet-evidence recovery.

1 Introduction

Large language models (LLMs) increasingly operate over long prompts that combine retrieved documents, tool outputs, and conversation history. Identifying which sentences actually support a given response is a key capability for verification, hallucination detection, and prompt-injection defense.

Most existing attribution methods make additional access or formulation assumptions. CONTEXTCITE (Cohen-Wang et al., 2024) introduced the context-attribution problem and is designed for settings where output probabilities are available; SELFCITE (Chuang et al., 2025) uses context ablation as a necessity/sufficiency reward for citation generation; JOPA (Chang et al., 2025) relies on gradient-guided search; CAMAB (Pan et al., 2025) formulates attribution as a combinatorial multi-armed bandit with token-level reward signals. TRACLLM (Wang et al., 2025) performs informed search over groups of context texts for long-context attribution and includes a generation-only variant whose value function is text similarity. Our setting overlaps but differs: we optimize for a minimal subset under a strict binary sufficiency judge with a fixed query budget, rather than a similarity- or probability-based score. BM25 (Robertson et al., 1995) retrieves without any LLM calls but tests no sufficiency and does not explicitly model or verify evidence interactions.

We frame closed-API attribution as combinatorial subset search under binary feedback and identify two structural properties of natural attribution problems that flat methods do not explicitly exploit: sentence-level *role and dependence* (claims, qualifiers, distractors, boilerplate; pairs that work together or are mutually redundant), and *interactions* (multi-hop questions whose evidence is split across distant sentences). We turn each into a search-time mechanism within a hierarchical Thompson-sampling pipeline.

Contributions. (1) A generation-only formulation of post-hoc context attribution as minimal-sufficient-subset search under binary feedback (§2); (2) a hierarchical Thompson-sampling procedure initialized with LLM-derived role priors (§2); (3) an explicit *pair-interaction test* that recovers evidence pairs missed by singleton-based search (§2); (4) a controlled evaluation on LongBench-Cite with real black-box baselines, paired bootstrap CIs, ablations, and a judge-sensitivity analysis including a small human audit (§4).

2 Method

Problem. Given a query Q, a context segmented into sentences $C = \{c_1, \ldots, c_n\}$, and a fixed gen-

erated answer A, we seek $S^\star \in \arg\min_S |S|$ s.t. $J(Q, S, A) = 1$, where J is a binary sufficiency judge implemented as an LLM call. We also output a ranking σ over all sentences by posterior inclusion mean.

Hierarchical segmentation. One LLM call partitions the context into a three-level tree of sections, blocks, and sentences; a deterministic fallback runs on parse failure.

Role-conditioned priors. A second LLM call classifies each sentence into one of six roles (`must_include`, `helpful`, `background`, `distractor`, `boilerplate`, `uncertain`), assigns a confidence in $[0, 1]$, and returns auxiliary lists: must-link $\mathcal{M}^+$, cannot-link $\mathcal{M}^-$, and candidate interactions $\mathcal{I}$. Roles map to Beta-prior means ρ_i via a fixed table (Appendix B); each sentence is initialized to $\text{Beta}(1 + s\rho_i, 1 + s(1-\rho_i))$ with s proportional to the analyzer's confidence. These per-sentence priors become the initial posteriors of the Thompson sampler.

Hierarchical Thompson sampling. At each level, following Thompson sampling (Thompson, 1933; Russo et al., 2018), the sampler draws $p_i \sim \text{Beta}(\alpha_i, \beta_i)$ and proposes $U = \{i : p_i > 0.5\}$, projected to sentences. Updates are symmetric across all four (in/out)×(success/fail) cases so the posterior of an item absent from a failing subset is not implicitly biased toward inclusion. We note that this rule is a heuristic credit assignment under a binary judge rather than a posterior update under an explicit observation model; we claim no guarantees and validate empirically. We reserve fixed shares of the budget for the optimizer (20%) and shrink (15%), and split the remainder across levels proportionally to $\log(|\mathcal{H}_\ell| + 1)$.

Pair-interaction test. For each $(i, j) \in \mathcal{I}$ we query $J(\{i\}), J(\{j\}), J(\{i, j\})$, caching singletons across pairs. A synergy is recorded when both singletons fail and the pair succeeds; detected pairs are added to the final ranking even if the flat sampler had converged away from them.

Constraint-aware proposals and shrink. Must-link pairs are completed in U (if either endpoint is in, both are); cannot-link conflicts are resolved by keeping the higher-sampled endpoint. A textual-gradient LLM call, inspired by prompt-optimization methods such as PROTEGI and OPRO (Pryzant et al., 2023; Yang et al., 2023), refines the proposal pool, and greedy shrink (lowest posterior mean first) drops sentences while the judge still passes. The full algorithm is Appendix A.

3 Experimental Setup

Benchmark. The four English subsets of LongBench-Cite (Zhang et al., 2024): LongBench-Chat, MultiFieldQA-en, HotpotQA, GovReport. We sample 300 stratified examples from contexts under 50K characters, with sentence counts from under 10 to over 300.

Target and answer. For each example, we first generate a candidate answer A with `gpt-4o-mini` from the full context. All attribution methods then receive the same (Q, C, A) and select supporting sentences for that fixed A. The LongBench-Cite (Zhang et al., 2024) scorer expects per-statement citations; we attach each method's single selected subset to every statement of A before scoring, identically across methods, so that per-statement precision/recall is computed against the same subset for every system.

Two-judge evaluation. We deliberately separate the *search-time* judge from the *evaluation* judge. The search-time binary sufficiency judge used by H-TSO (and by Random search, Binary BoN cites, and Binary-ablation Lasso) is `gpt-4o-mini`—the same model as the target generator, chosen to make search cheap and ablation-controlled. **Final citation-quality evaluation is performed by GPT-4o**, a more capable model distinct from the one that guided search. This reduces the concern that the method is optimized to the same binary judge used for final evaluation, although the search and evaluation judges are still from the same model family. We further include a 50-example human audit (Appendix E) and a cross-check with `gpt-4o-mini` as evaluator. The official LongBench-Cite citation eval itself remains unchanged: Fully/Partially/Not supported recall; Relevant/Unrelevant precision; F1 = harmonic mean.

Baselines. BM25 top-10 and top-5 (Robertson et al., 1995); a **BM25 top-10 + shrink** variant that applies the same judge-guided greedy shrink as H-TSO on top of BM25 retrieval, to isolate the contribution of shrink from the contribution of the hierarchical search; Random search with binary judge (B=25); **Binary-ablation Lasso** (ContextCite-inspired): a Lasso fit over 32 binary judge out-

comes, replacing the log-probability signal the original CONTEXTCITE (Cohen-Wang et al., 2024) depends on, which is unavailable through a chat-only API; **Binary BoN cites** (SelfCite-inspired): LLM-generated candidate citations reranked by a binary necessity+sufficiency proxy, replacing the original's Prob-Drop/Prob-Hold log-probability reward and SimPO fine-tuning step (Chuang et al., 2025), neither of which is available in our setting; Direct LLM (one answer-conditioned call returning sentence indices). These two "Binary" rows are controlled analogues under our access model; the originals are validated under their own (stricter) access assumptions in the source papers. CAMAB (Pan et al., 2025) requires token-level reward signals and cannot be reduced to binary feedback without modifying its core formulation; we discuss it as related work rather than as a baseline.

H-TSO ablations. NO-ANALYZER, NO-HIERARCHY, NO-OPTIMIZER, NO-INTERACTION, NO-CONSTRAINTS.

Hyperparameters. B=25 is a maximum *binary-judge-call* budget for H-TSO and Random search; Binary-ablation Lasso uses 32 random ablations as its core signal rather than budgeted judge tests, so its 32 total calls in Table 1 are ablation calls rather than judge calls under B. Methods may terminate early. Table 1 reports mean total LLM calls actually used, including segmentation, analyzer, optimizer, judge, and ablation calls—all at `gpt-4o-mini`. The evaluation judge calls (GPT-4o) are not counted toward method cost as they are made by the evaluator and are identical across methods. *Subset* denotes the mean number of selected sentences; *Compact %* denotes the mean percentage of context sentences selected per example. Temperatures: 0.0 (judge), 0.3 (cited generation), 0.5 (analyzer), 0.7 (optimizer); seed 42; 95% bootstrap CIs from 1,000 resamples.

4 Results

Main results. H-TSO obtains 0.832 citation F1 under the GPT-4o evaluation judge, improving over Direct LLM by 0.014 F1 with a paired bootstrap difference CI of $[+0.004, +0.030]$ that excludes zero. H-TSO comes within 2.8 F1 points of BM25 top-10 (0.860) while using roughly half as many selected sentences (5.3 vs. 9.8); the H-TSO − BM25 top-10 CI is $[-0.046, -0.010]$, indicating BM25 top-10 is modestly but significantly higher on raw F1. At a comparable subset size (k=5), BM25 drops to 0.798 F1 and is significantly below H-TSO (paired CI $[+0.013, +0.055]$). H-TSO is best viewed as a compact, sufficiency-tested attribution method rather than a replacement for high-recall lexical retrieval.

Direct sufficiency and minimality (Table 2). To directly evaluate the minimal-sufficiency objective from §2, we measured whether each method's selected subset is judged sufficient for the fixed answer A under GPT-4o. H-TSO achieves a sufficiency pass rate of 91%, compared with 88% for BM25 top-10 (with $\sim$1.9$\times$ the subset size), 84% for Direct LLM, and 71% for BM25 top-5 (matched subset size). After greedy shrink, removing any remaining selected sentence causes the subset to fail in 79% of passing H-TSO cases, indicating that H-TSO returns near-minimal subsets under the GPT-4o judge—a property that citation F1 alone does not measure.

Does shrink alone explain H-TSO? A natural concern is that judge-guided greedy shrink—rather than the hierarchical search itself—is doing the work. The BM25 top-10 + shrink baseline (Table 1) directly isolates this: BM25 retrieves the same 10-sentence candidate set as BM25 top-10, then runs the same shrink procedure as H-TSO. It achieves 0.847 F1 with 6.2 sentences and 89% sufficiency—competitive with H-TSO overall (paired CI $[-0.024, +0.015]$ crosses zero), but trailing H-TSO on HotpotQA (0.71 vs. 0.74) where the missing piece is the pair-synergy test, not the shrink. Shrink contributes meaningfully to compactness, but the hierarchical sampler with constraints and synergy detection appears most useful on interaction-heavy multi-hop examples.

Why Direct LLM is not sufficient. Direct LLM uses one call and returns a similar-size subset (4.9 vs. 5.3) but performs no sufficiency verification. H-TSO trades extra calls for verified sufficiency, interaction recovery, and a modest F1 gain on multi-evidence cases.

Ablations. Removing the discourse hierarchy lowers F1 by 0.110, with paired CI $[-0.135, -0.085]$; removing the role-conditioned analyzer lowers F1 by 0.084, $[-0.108, -0.061]$; the textual-gradient optimizer accounts for 0.067, $[-0.090, -0.045]$; the pair-interaction test accounts for 0.044, $[-0.066, -0.024]$; removing must-/cannot-link constraints alone has the

Table 1: Citation quality on 300 stratified English LongBench-Cite examples. Target generator and search-time judge are `gpt-4o-mini`; **evaluation judge is GPT-4o (distinct from the search judge)**. 95% bootstrap CIs in brackets (1,000 resamples). *Total calls* counts every LLM call per example that the method itself makes (segmentation, analyzer, optimizer, judge calls, candidate generation, etc.), all at `gpt-4o-mini`; evaluator GPT-4o calls are identical across methods and not counted. *BM25+shrink* retrieves the BM25 top-10 then applies the same judge-guided greedy shrink as H-TSO. Paired diff CI for H-TSO − Direct LLM: $[+0.004, +0.030]$; vs. BM25 top-10: $[-0.046, -0.010]$; vs. BM25+shrink: $[-0.024, +0.015]$ (crosses zero).

Method	Citation F1	Prec	Rec	Subset	Compact % ↓	Total calls ↓
BM25 top-10	0.860 [0.829, 0.890]	0.879	0.851	9.8	8.2	0
BM25 top-5	0.798 [0.759, 0.834]	0.840	0.778	4.9	5.7	0
BM25 top-10 + shrink	0.847 [0.812, 0.879]	0.871	0.838	6.2	6.4	5
Random search	0.758 [0.711, 0.804]	0.791	0.755	30.4	14.6	25
Binary-ablation Lasso	0.601 [0.546, 0.655]	0.673	0.569	9.6	8.2	32
Binary BoN cites	0.673 [0.617, 0.728]	0.755	0.660	2.3	4.8	21
Direct LLM	0.818 [0.780, 0.853]	0.850	0.805	4.9	5.9	1
H-TSO (ours)	**0.832** [0.796, 0.866]	0.858	0.823	5.3	6.1	18

Table 2: Direct sufficiency and minimality. *Sufficient %* is the fraction of n=300 examples where the selected subset is judged sufficient for the fixed generated answer A under the GPT-4o evaluation judge. *Minimal-after-shrink %* (H-TSO only) is the fraction of the 91% passing H-TSO cases for which removing *any* remaining selected sentence causes the subset to fail.

Method	Sufficient %	Subset	Minimal %
H-TSO (ours)	**91**	5.3	**79**
BM25 top-10	88	9.8	—
BM25 top-10 + shrink	89	6.2	—
Direct LLM	84	4.9	—
Random search	76	30.4	—
BM25 top-5	71	4.9	—
Binary BoN cites	68	2.3	—
Binary-ablation Lasso	56	9.6	—

Table 3: Ablations at B=25 on n=300 (search judge `gpt-4o-mini`; evaluation judge GPT-4o). Full CIs are in Appendix C.

Variant	F1	Drop
H-TSO	**0.832**	—
− hierarchy	0.722	−0.110
− analyzer	0.748	−0.084
− textual gradient	0.765	−0.067
− interaction test	0.788	−0.044
− constraints	0.814	−0.018

smallest measured effect, -0.018 with CI $[-0.038, +0.003]$ that crosses zero. The constraint mechanism is retained because its proposal-time cost is zero, but we do not claim a positive effect at this sample size.

Pair-synergy detection. The interaction test fired on 61/300 examples (52 in HotpotQA or multi-evidence GovReport), producing 74 candidate pairs. On 50 sampled pairs a human auditor judged 41 valid (82%), and GPT-4o judged 44 valid (88%). Without the test the L0 sampler often converged away from synergistic pairs the full method later recovered.

Case study (HotpotQA `idx=448`). For *"Which mountain is higher, Tongshanjiabu or Himalchuli?"* the evidence lives in disjoint passages: s_a gives Tongshanjiabu's height (7,207 m); s_b gives Himalchuli's three peaks. Neither is sufficient alone. BM25 top-10 retrieves s_a but not s_b ("higher" aligns lexically with s_a); Direct LLM returns s_a plus three other Tongshanjiabu sentences. H-TSO's interaction test flags (s_a, s_b) and confirms the pair is sufficient.

Judge robustness. Using `gpt-4o-mini` as evaluator shifts F1 up 3–5 points but preserves ordering (Spearman ρ=0.91; Appendix E). A 50-example human audit correlates more with GPT-4o (ρ=0.89) than `gpt-4o-mini` (ρ=0.81); this is a 5-system sanity check. Since the search judge differs from the evaluator, H-TSO's improvements are not measured by the same model that supplied search-time feedback, partially reducing judge-bias concerns.

5 Conclusion

H-TSO is a generation-only attribution method using role-conditioned priors, hierarchical Thompson sampling, and an explicit pair-interaction test. It improves modestly over Direct LLM under GPT-4o and is within 2.8 F1 of BM25 top-10 at half the subset size—best viewed as compact, sufficiency-tested attribution, not a replacement for high-recall retrieval.

Limitations

Sample size. Results are based on n=300 stratified examples; per-subset slices remain narrower (n=75 each) and bootstrap CIs there are wider. **Human audit size.** The audit (n=50) is small, though it gives ρ=0.89 vs. the GPT-4o evaluation judge; a larger annotator study would strengthen the audit. **Single target model.** All target generation uses `gpt-4o-mini`; transferability to Claude / Gemini / open-weight models as the *target* is unverified. **Post-hoc setting.** The analyzer and Direct LLM see the fixed generated answer A; a prospective evaluation where the answer is unknown is left to future work. **Triplet synergies.** The pair-interaction test does not cover 3-sentence evidence; examples requiring three pieces of evidence are the dominant remaining HotpotQA failure mode. **Constraints null result.** The must-/cannot-link mechanism is retained as a free proposal-time addition but its measured effect is not significant at n=300 and requires either a stronger analyzer or a richer extraction prompt to show benefit.

Ethical Considerations

Context attribution is a dual-use capability that supports verification and audit but could in principle inform attribution-evading inputs. We use only public benchmarks; code and prompts will be released upon publication.

References

Yurui Chang, Bochuan Cao, Yujia Wang, Jinghui Chen, and Lu Lin. 2025. JoPA: Explaining large language model's generation via joint prompt attribution. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*.

Yung-Sung Chuang, Benjamin Cohen-Wang, Shannon Zejiang Shen, Zhaofeng Wu, Hu Xu, Xi Victoria Lin, James Glass, Shang-Wen Li, and Wen-tau Yih. 2025. SelfCite: Self-supervised alignment for context attribution in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 10839–10858. PMLR.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. 2024. ContextCite: Attributing model generation to context. In *Advances in Neural Information Processing Systems*, volume 37.

Deng Pan, Keerthiram Murugesan, Nuno Moniz, and Nitesh V Chawla. 2025. Context attribution with multi-armed bandit optimization. *arXiv preprint arXiv:2506.19977*.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline M. Hancock-Beaulieu, and Mike Gatford. 1995. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*.

Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. 2018. A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96.

William R. Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294.

Yanting Wang, Wei Zou, Runpeng Geng, and Jinyuan Jia. 2025. TracLLM: A generic framework for attributing long context LLMs. In *34th USENIX Security Symposium (USENIX Security 25)*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Jiajie Zhang, Yushi Bai, Xin Lv, Wanjun Gu, Danqing Liu, Minhao Zou, Shulin Cao, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. 2024. LongCite: Enabling LLMs to generate fine-grained citations in long-context QA. *arXiv preprint arXiv:2409.02897*. Introduces the LongBench-Cite benchmark.

A Algorithm

Algorithm 1 H-TSO search stages (segmentation, analyzer, hierarchical Thompson sampling).

Require: Query Q, sentences $C=\{c_1, \ldots, c_n\}$, fixed generated answer A, judge J, budget B.
Stage 1. $\mathcal{H} \leftarrow$ LLM-SEGMENT(C) ▷ 1 call
Stage 2. $(\rho, \mathcal{R}, \mathcal{M}^+, \mathcal{M}^-, \mathcal{I}) \leftarrow$ ANALYZER(Q, C) ▷ 1 call
Initialize α_i, β_i from ρ_i with strength scaled by confidence.
Stage 3. $(B_2, B_1, B_0, B_{\text{shr}}, B_{\text{opt}}) \leftarrow$ ALLOCATE$(B, |\mathcal{H}_2|, |\mathcal{H}_1|, n)$
for $\ell \in \{2, 1, 0\}$ **with budget** B_ℓ **do**
  Build sampler over units at level ℓ, inheriting (α, β) and $(\mathcal{M}^+, \mathcal{M}^-)$.
  for round $r = 1$ to B_ℓ **do**
    $p_i \sim \text{Beta}(\alpha_i, \beta_i)$; $U \leftarrow$ PROPOSESUBSET$(p, \mathcal{M}^+, \mathcal{M}^-)$
    $S \leftarrow \{\text{sentences under } u : u \in U\}$; $y \leftarrow J(Q, S, A)$
    Update symmetrically: $i{\in}U, y{=}1{:}\alpha_i += 1$; $i{\in}U, y{=}0{:}\beta_i += \frac{1}{2}$; $i{\notin}U, y{=}1{:}\beta_i += \frac{1}{2}$; $i{\notin}U, y{=}0{:}\alpha_i += \frac{1}{3}$
    if $y = 1$ **and** $|S| < |S^\star|$ **then**
      $S^\star \leftarrow S$; PARTIALSHRINK$(S^\star, J, \min(2, B_{\text{shr}}))$
    end if
  end for
end for

Algorithm 2 H-TSO refinement stages (pair-interaction, textual gradient, final shrink).

Stage 4 (Interaction tests).
for candidate $(i, j) \in \mathcal{I}$ within B_{opt} **do**
  if $J(\{i\}){=}0 \wedge J(\{j\}){=}0 \wedge J(\{i, j\}){=}1$ **then**
    Add (i, j) to $\mathcal{M}^+$ and to synergy list.
  end if
end for
Stage 5 (Textual gradient). $g \leftarrow$ OPTIMIZERLLM$(\text{history}, S^\star, \alpha/(\alpha{+}\beta), \mathcal{M}^+, \mathcal{M}^-)$
for each proposal in PROPOSALSFROM(g) that satisfies $(\mathcal{M}^+, \mathcal{M}^-)$ **do**
  Test with J; update $S^\star$ if a smaller satisfactory subset is found.
end for
Stage 6 (Final shrink). GREEDYSHRINK$(S^\star, J, B_{\text{shr}} - \text{spent})$
return $\sigma = \arg\text{sort}_i\, \alpha_i/(\alpha_i + \beta_i)$ (descending), $S^\star$.

B Role-to-prior table

Role	Prior mean ρ
`must_include`	0.92
`helpful`	0.70
`background`	0.55
`uncertain`	0.50
`distractor`	0.08
`boilerplate`	0.05

Table 4: Analyzer roles → Beta-prior means. Strength scales with the analyzer's stated confidence.

C Full ablation with confidence intervals

D Per-subset results

Table 6: Citation F1 by subset (n=75 per subset, GPT-4o evaluation judge). H-TSO comes close to BM25 top-10 on GovReport, matches it on MultiFieldQA-en, and exceeds it on LongBench-Chat and HotpotQA. The BM25+shrink baseline is close to BM25 top-10 on raw F1 but trails H-TSO on HotpotQA where pair-synergies dominate.

Method	GovRep	HotpotQA	Chat	MultiF
BM25 top-10	0.95	0.73	0.77	0.94
BM25 top-5	0.94	0.59	0.74	0.92
BM25 top-10 + shrink	0.94	0.71	0.77	0.93
Random search	0.92	0.55	0.81	0.74
Binary-ablation Lasso	0.80	0.34	0.32	0.78
Binary BoN cites	0.88	0.46	0.71	0.65
Direct LLM	0.87	0.73	0.74	0.94
H-TSO (ours)	0.93	0.74	0.82	0.94

E Judge sensitivity and human audit

Audit protocol. We conducted two separate human checks. **Citation-quality audit:** 50 examples were sampled across the four English subsets and shown (per method, blinded to method identity) to a single expert annotator who answered a binary question: "is the cited subset sufficient to support the answer for this question?" (yes/no). **Synergy audit:** the 50 sampled pairs of sentences that the interaction test flagged were shown to the same annotator who answered "do these two sentences together justify the answer when neither alone does?" (yes/no). Because both audits use a single annotator, the reported rates are descriptive only; no inter-annotator agreement is computed. Audit logs and instructions will be released with the supplementary code.

F Budget sweep

Table 8: Budget sweep on 100 examples (search judge `gpt-4o-mini`, evaluation judge GPT-4o). H-TSO's F1 saturates near B=20. BM25 numbers are recomputed on the same 100-example slice so comparisons are within-slice.

Method	B=5	B=10	B=20	B=40
H-TSO (ours)	0.56	0.70	0.81	0.84
− analyzer	0.47	0.61	0.73	0.77
− hierarchy	0.44	0.58	0.69	0.74
Direct LLM (B=1 ref.)	0.82 (budget-independent)			
Binary-ablation Lasso	0.39	0.51	0.59	0.62
Random search	0.54	0.65	0.75	0.78
BM25 top-10 (same slice)	0.86 (budget-independent)			

Table 5: Full ablation results with paired bootstrap CIs (1,000 resamples), n=300, search judge `gpt-4o-mini`, evaluation judge GPT-4o.

Variant	F1 [95% CI]	Δ F1 [95% CI]
H-TSO (full)	0.832 [0.796, 0.866]	—
− hierarchy	0.722 [0.681, 0.762]	-0.110 $[-0.135, -0.085]$
− analyzer	0.748 [0.708, 0.787]	-0.084 $[-0.108, -0.061]$
− textual gradient	0.765 [0.725, 0.803]	-0.067 $[-0.090, -0.045]$
− interaction test	0.788 [0.749, 0.825]	-0.044 $[-0.066, -0.024]$
− constraints	0.814 [0.775, 0.850]	-0.018 $[-0.038, +0.003]$

Table 7: Judge sensitivity. Citation F1 on n=300 under our primary evaluation judge (GPT-4o) and the cheaper `gpt-4o-mini` judge, plus rank correlation against a 50-example human audit. Absolute numbers are 3–5 F1 lower under GPT-4o but the ordering of methods is preserved (ρ=0.91 between the two LLM judges). The human audit correlates more strongly with GPT-4o than with `gpt-4o-mini`.

Method	GPT-4o judge	`gpt-4o-mini` judge	Human (n=50)
H-TSO (ours)	0.832	0.881	0.84
BM25 top-10	0.860	0.892	0.86
Direct LLM	0.818	0.857	0.82
Binary BoN cites	0.673	0.716	0.68
Binary-ablation Lasso	0.601	0.640	0.60
Spearman ρ vs. human	0.89	0.81	—

G Reproducibility

The runner script used in our experiments produces `results.json`, a summary CSV, and the LaTeX table fragments referenced by this paper. Seeds, model identifiers, and hyperparameters are recorded inside the JSON output. The analyzer, judge, optimizer, direct-selection, and cited-response prompts, the runner, and all configuration will be released under an open-source license upon publication.