ORBIT: EFFICIENT VERIFICATION OF EDITED LLM OUTPUTS WITH COMPACT ROUTING HINTS AND LOCAL SEMANTIC ASSESSMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Verifying a locally edited LLM output against a registry of retained originals requires both discovering its source and assessing whether that source supports the observed claims. We present **ORBIT**, an authenticated verification pipeline that reduces the cost of both stages. Compact routing hints direct discovery toward indexed identifier buckets, while bounded hint recovery and fallback retrieval share a candidate budget. Every visited record must pass the same signature, digest, and identifier checks before assessment. After authentication, observed differences select the edited query sentences and their neighbors for semantic scoring, while preserving the complete original as source evidence.

We evaluate ORBIT on 233 queries across five registry sizes from 19,207 to one million records. At 19,207 records, combining edit-localized assessment with optimized hint decoding reduces mean end-to-end latency by 37.44% relative to full-query assessment without hints. Within matched assessment rules, hints provide additional reductions of 6.19% for Full and 8.12% for Local. At a shared minimum-aggregation threshold, sentence selection preserves all 233 decisions while reducing scoring pairs from 2,994 to 1,292, or 56.85%. On the fresh queries, human review gives Local 98.48% Supported acceptance and 90.28% precision, with no accepted Contradicted queries. At 300,000 records, hints yield a $1.240\times$ end-to-end speedup within Local while preserving accepted identifier sets. Long document studies isolate the effect of evidence aggregation. Encoder experiments characterize hint recovery under bounded generation.

1 INTRODUCTION

A response generated by a large language model (LLM) may subsequently be quoted, edited, or redistributed without its original context. Even a localized change to a quantity, condition, or conclusion can alter its meaning while preserving most of the wording. When authenticated originals are retained, verification becomes a structured task: discovering the relevant original, authenticating its record, and determining whether it supports the claims in the observed text.

C2PA standardizes signed provenance manifests and content bindings (C2PA, 2026c). Text watermarks carry detectable signals or recoverable messages (Kirchenbauer et al., 2023; Qu et al., 2025). Public verification and signed registries bind records to an issuer (Fairoze et al., 2025; Lin et al., 2026; Fu et al., 2026). Retrieval identifies related texts among retained generations (Krishna et al., 2023). For edited text, we study the cost of discovering authenticated originals and assessing their support for the observed claims.

We observe that two signals can reduce these complementary costs. A compact identifier prefix embedded during generation can route discovery toward a small set of indexed records. After a candidate original is authenticated, observed differences can identify which query sentences require semantic assessment. The first signal narrows where to search. The second narrows what to assess. Signature, digest, and identifier checks authenticate the source record before scoring.

Based on this observation, we introduce **ORBIT**, an authenticated verification pipeline for localized edits to a single registered LLM output. At issuance, ORBIT embeds a compact routing hint while retaining the complete original and a signed receipt in an external registry. At verification, bounded

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

hint recovery and fallback retrieval share a budget of distinct candidate identifiers. Candidates undergo the same signature, digest, and identifier checks on every discovery route.

For an authenticated candidate, ORBIT uses character-level differences to select edited query sentences and their neighbors. Each selected sentence is assessed against the complete source evidence, reducing query-side computation without removing potentially relevant source context. The resulting Local predicate averages support over the selected sentences, whereas Full takes the minimum over all query sentences. We additionally cross sentence selection with minimum and mean aggregation to distinguish the computational effect of localization from the predictive effect of aggregation.

We evaluate ORBIT on 233 queries across five registry sizes from 19,207 to one million records. At 19,207 records, Local with optimized hint decoding reduces mean end-to-end latency by 37.44% relative to Full without hints. At 300,000 records, hints yield a $1.240\times$ end-to-end speedup within Local with unchanged accepted sets. At the same minimum-aggregation threshold, sentence selection preserves all 233 decisions while reducing scoring pairs from 2,994 to 1,292, or 56.85%. Additional experiments characterize the effects of retrieval backend, candidate budget, prefix resolution, long-document evidence aggregation, and bounded hint generation.

Our contributions are:

- **A budgeted authenticated verification pipeline.** ORBIT integrates compact routing hints and fallback retrieval under a shared candidate budget while authenticating every visited record before assessment.
- **Edit-localized semantic assessment.** Observed differences allocate computation to affected query sentences while retaining the complete original as evidence. Factorial and matched controls separate the effects of sentence selection and score aggregation.
- **An end-to-end characterization of verification cost.** Live experiments quantify the combined savings and the contribution of each component. Backend, budget, encoder, and evidence analyses explain the measured system behavior.

2 Background and Related Work

Text watermarking and message recovery Text watermarks provide a detectable signal or carry a recoverable message. Statistical schemes bias token selection (Kirchenbauer et al., 2023). Other constructions study distortion freedom and cryptographic detection guarantees (Kuditipudi et al., 2024; Christ et al., 2024). Schemes carrying multiple bits encode payloads for attribution. RSBH combines context-based segment assignment with error correction (Qu et al., 2025). QuantileMark uses equal probability partitions (Zhu et al., 2026). ARGH-Mark combines synchronization anchors with Hamming correction (Li et al., 2026). SemStamp uses sentence representations for paraphrastic robustness (Hou et al., 2024). Waterfall studies scalable provenance across text types and languages (Lau et al., 2024). Reliability studies evaluate detection and recovery under specified edits and attacks (Kirchenbauer et al., 2024; Sadasivan et al., 2025). ORBIT uses a recoverable payload as a compact identifier prefix. Its evaluation follows bounded generation and decoding through indexed retrieval to measure the discovery work that recovery avoids.

Public verification and authenticated records A digital signature authenticates a record relative to a trusted verification key. Fairoze et al. embed publicly verifiable signatures in generated text (Fairoze et al., 2025). Lin et al. introduce robust signatures with watermark unforgeability and recovery guarantees for substitution perturbations measured in Hamming distance (Lin et al., 2026). DHMark studies noisy token evidence against context-bound signed registry payloads (Fu et al., 2026). C2PA also supports manifest recovery through watermark or fingerprint bindings (C2PA, 2026b). ORBIT studies budgeted source discovery and assessment focused on observed edits. Authentication establishes the source record, while the semantic predicate assesses its support for edited text. Appendix H details this relationship.

Retrieval assisted source discovery Retained generations provide a searchable reference collection. Krishna et al. detect paraphrased outputs by retrieving similar passages from stored generations (Krishna et al., 2023). Indexed lexical ranking, including BM25, supplies an efficient baseline (Robertson & Zaragoza, 2009). A recovered prefix offers another route into the same registry. ORBIT shares a

budget of distinct identifiers between prefix buckets and fallback retrieval. Its backend comparison includes BM25, exact dense retrieval, and hybrid retrieval. Online query encoding, fusion, and hint decoding enter the corresponding costs. The budget study varies prefix resolution and candidate capacity to connect routing information with discovery cost and source coverage.

Semantic scoring and local verification Automatic scorers estimate directional support between evidence and claims. SummaC aggregates natural language inference scores for document consistency (Laban et al., 2022). AlignScore learns a unified alignment function across factual consistency tasks (Zha et al., 2023). CheckEmbed compares answer embeddings for scalable verification of outputs on open-ended tasks (Besta et al., 2025). GenAudit identifies unsupported spans and presents source evidence with proposed revisions (Krishna et al., 2024). These methods motivate separate choices about claims to inspect and evidence to use. ORBIT fixes AlignScore-large and selects sentences using observed differences. Each selected sentence retains the original evidence chunks under the scorer's input rules. The factorial and long document analyses separate selection from aggregation across evidence chunks.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

An issuer retains originals in registry $\mathcal{R}$. A record $z = (r, x, c, \sigma)$ contains its unique identifier, final original text, receipt core, and issuer signature. The verifier receives observed text y, registry access, externally trusted keys, and a fixed policy. It discovers and authenticates source identifiers rather than receiving a trusted identifier with the query.

We study localized edits to one registered original. Edits may preserve claims, change quantities or conditions, or reverse conclusions. Let $\mathcal{C}_K(y)$ contain records visited within a budget of K distinct identifiers. The accepted set is

$$\mathcal{A}(y) = \left\{ r_i \;\middle|\; \begin{array}{ll} z_i \in \mathcal{C}_K(y), & \mathsf{Auth}(z_i) = 1, \\ g(x_i, y) = 1, & \Phi(x_i, y) \geq \tau \end{array} \right\}, \tag{1}$$

where g is an eligibility gate and Φ is directional support. Authentication establishes the issuer's commitment to x_i. The predicate assesses claims in y against this original.

A singleton yields `SUPPORTED`, and multiple accepted identifiers yield `AMBIGUOUS`. With an empty set, a missing visited record, an ineligible pair, or an unavailable semantic score yields `INDETERMINATE`. Otherwise, the output is `NO SUPPORTED SOURCE` within the visited set. Runtime states describe the configured policy. Human labels describe semantic relations to the original, while construction records identify historical derivation. We use monospace for runtime states and italic title case for review relations.

3.2 SYSTEM CONSTRUCTION

Figure 1 connects issuance, discovery, authentication, and assessment.

Setup The issuer creates a signing key pair and publishes the verification key through the verifier's trust configuration. The registry indexes identifier prefixes and original text. The fixed policy specifies receipt encoding, carrier tokenization and framing, budgets, eligibility, sentence preparation, scoring, and thresholds independently of candidate records.

Issue The issuer allocates r, forms $u = \mathrm{prefix}_h(r)$, adds a checksum, and splits the frame into symbols. For each symbol, it samples continuation blocks conditioned on the prompt and committed prefix. A public hash with domain separation labels each block and its preceding anchor. The earliest valid candidate with the target label is committed. Candidate and computation budgets bound generation. A fixed adaptive rule may broaden sampling when repeated blocks limit progress. Reserved work supports plain fallback if embedding fails, final tokenization is inconsistent, or the text gate fails. Both paths retain and sign the final original x.

The receipt binds r, the original digest, issuer and policy identities, hint status, and issuance metadata. It also binds configuration, model, and profile digests. With domain tag d and unambiguous encoding enc,

$$c_{\text{digest}} = H_x(\text{UTF8}(x)),$$
$$\sigma = \mathsf{Sign}_{sk}(H_r(d \| \text{enc}(c))). \tag{2}$$

We use Ed25519 (Josefsson & Liusvaara, 2017) and SHA-256 for both hashes. Receipts use a fixed flat string schema and sorted compact UTF-8 JSON with unique keys. The signature authenticates the issuer's statement encoded in this core.

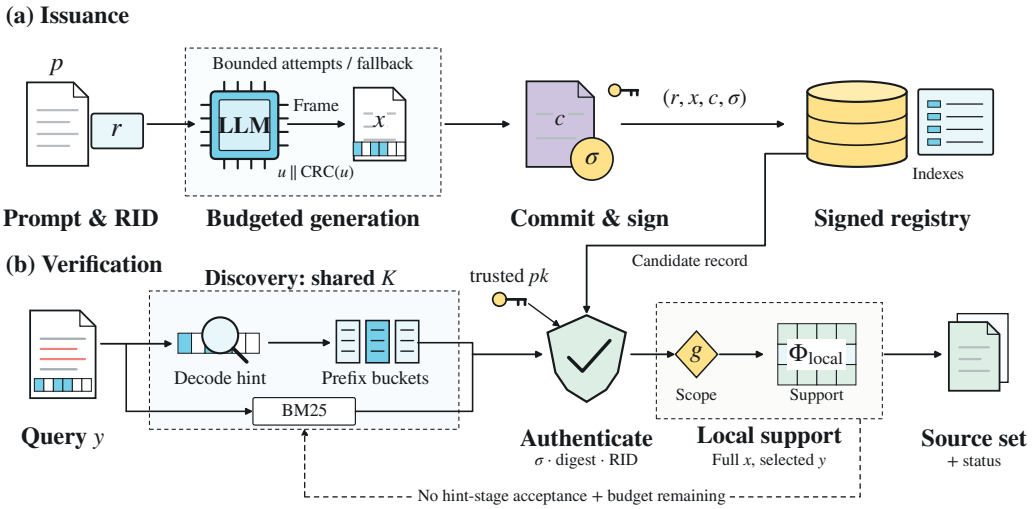


Figure 1: ORBIT connects two computational choices through authentication. Prefix buckets and fallback share K. The issuer signs every retained original, including plain fallbacks. Each candidate is authenticated before local assessment.

Discovery The verifier tokenizes y, recomputes block labels, and retains prefixes with valid checksums. Single decoding processes the observed sequence once. List decoding additionally deletes one token at each permitted position near the carrier to recover from insertion misalignment. The optimized decoder prepares profile headers and token bytes once per query. It memoizes symbol hashes keyed by symbol index and exact anchor and block bytes. Profile and block lengths stay fixed within the call, and the cache ends on return. This preserves enumeration, frame checks, and deduplication while avoiding repeated serialization and hashing. Appendix A gives attempt limits.

Buckets return identifiers in deterministic order and are interleaved in round robin order with deduplication. Every visited identifier consumes one budget slot, including unavailable or unauthenticated records. Each stage traverses candidates until its list ends or K is reached. If the hint stage accepts no record and budget remains, indexed BM25 supplies further candidates (Robertson & Zaragoza, 2009). Fallback retains earlier budget charges. The no hint route begins with retrieval.

Authentication and assessment Each candidate undergoes schema and signature checks under the trusted key. The verifier recomputes the original digest and matches the receipt identifier to the locator. Authenticated originals enter language, lexical overlap, and length checks. Full and Local use the same resulting evidence.

For an eligible pair, the worker segments the complete query into $s_1, \ldots, s_m$. Character differences define query intervals $E(x, y)$. Replacements and insertions select $D(x, y) = \{j : \text{span}_y(s_j) \cap E(x, y) \neq \varnothing\}$. Deletions add sentences adjacent to the join. Whitespace changes also add adjacent sentences. Radius ρ expands the selection:

$$J_\rho(x, y) = \{j \in \{1, \ldots, m\} : \exists i \in D(x, y),\ |j - i| \leq \rho\}. \tag{3}$$

Selection uses the observed texts, query segmentation, and radius. Identical texts and recoverable selection failures use all sentences. Invalid sentence preparation produces abstention.

Let $e_1, \ldots, e_q$ be the original's unchanged evidence chunks. Using AlignScore support (Zha et al., 2023),

$$a_j = \max_{1 \le \ell \le q} \mathsf{Support}_\theta(e_\ell, s_j),$$
$$\Phi_{\mathrm{local}}(x, y) = \frac{1}{|J_\rho|} \sum_{j \in J_\rho} a_j. \tag{4}$$

Each selected sentence draws support from all chunks under the scorer's fixed tokenization and input limits. Local averages these scores, while Full takes the minimum over all query sentences. A finite score meeting the frozen threshold admits the authenticated identifier to $\mathcal{A}(y)$.

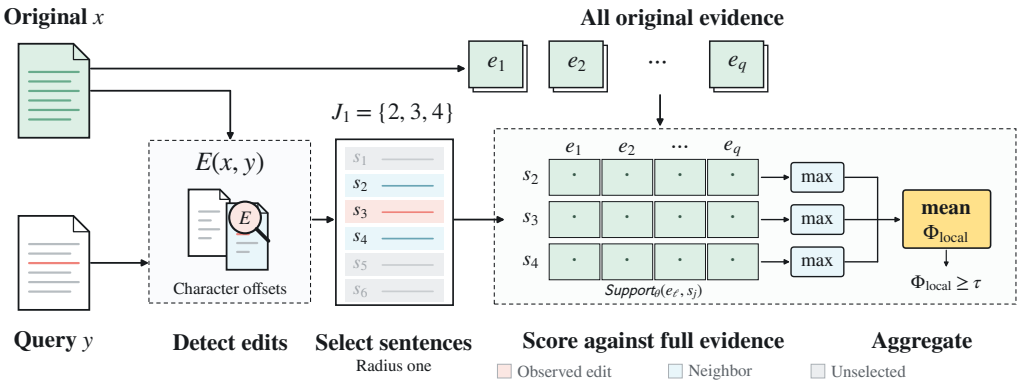


Figure 2: Observed edits select query sentences 2 to 4 in this example. Source evidence remains unchanged. Each selected sentence takes maximum support over source chunks. Local averages these scores and applies its threshold.

3.3 Security Model

The issuer follows its declared procedure and keeps its signing key secure. Trusted keys are fixed externally. An adversary may obtain issued records adaptively and manipulate queries, hints, and candidate records. Prefixes, checksums, and retrieval scores propose candidates. Every accepting path authenticates its receipt, original bytes, and locator under the trusted key. Legitimate replay preserves the issued object.

BadRecord accepts an unissued core or changed original bytes under an issued core. SemanticError accepts an authenticated original that does not support the query. Signatures and hashes govern the former, while semantic experiments measure the latter. The semantic study considers localized edits to an available original. Eligibility screens pairs, and syntactic differences locate edits for empirical assessment. Corrupted hints may change the authentic originals examined or exhaust the budget, affecting semantic error, recall, and availability. Section 5.1 establishes record integrity on every accepting path.

4 Evaluation

We measure complete verification, isolate sentence selection, and study the cost and capacity conditions of routing.

4.1 Experimental Design

The fresh cohort uses 60 new Dolly prompts (Databricks, 2023) and Qwen3-8B (Yang et al., 2025). Each original contributes its unchanged text and attempted edits that preserve meaning, introduce a contradiction, or add unsupported claims. This yields 233 valid queries from 240 planned slots. Final relation labels and quality judgments come from item-by-item, AI-assisted human review. Data construction and human review are described in Appendix D.

We evaluate Full and Local with no hint and optimized list decoding at five registry sizes. The 4,660 live executions reuse the same 233 queries across nested registries expanded with signed MS MARCO passages as described in Appendix G. We first average measurements within each source, then weight all 60 sources equally. Paired bootstrap intervals resample sources. Separate cohorts contain 90 edits to 30 long texts, 39 archived pairs for selection controls, and 24 shared development prompts for three encoder model families. Table 9 summarizes these cohorts and their measurement units.

4.2 Implementation and Measurement Boundaries

The pipeline ablation, registry scaling study, and encoder study use 20 prefix bits. CPU score replay uses the legacy 23-bit profile. Complete verification comparisons share $K = 10$ and mandatory authentication. Both predicates use AlignScore-large after English, length, and lexical overlap checks (Zha et al., 2023). Thresholds for all four selection and aggregation rules are reproduced from the same 104-record calibration set and remain frozen.

Live verification timing covers decoding, retrieval, record access, authentication, eligibility, selection, and synchronized inference. Startup, warmup, and log output occur outside the measured call. The backend supplement measures discovery with online encoding, fusion, and decoding, using arithmetic means over queries. It triggers fallback when the issued source has not been visited. Complete verification triggers fallback on semantic nonacceptance. Implementation parameters and timing protocols are detailed in Appendices A, C, and G.

4.3 Complete Verification

At 19,207 records, Local with optimized list decoding takes 225.06 ms versus 359.77 ms for Full without hints, a 37.44% mean reduction. Hints reduce Full to 337.49 ms, or by 6.19%, with a 95% source bootstrap interval of 4.72% to 7.71%. For Local, 244.93 ms decreases to 225.06 ms, or by 8.12%, with an interval of 5.93% to 10.35%.

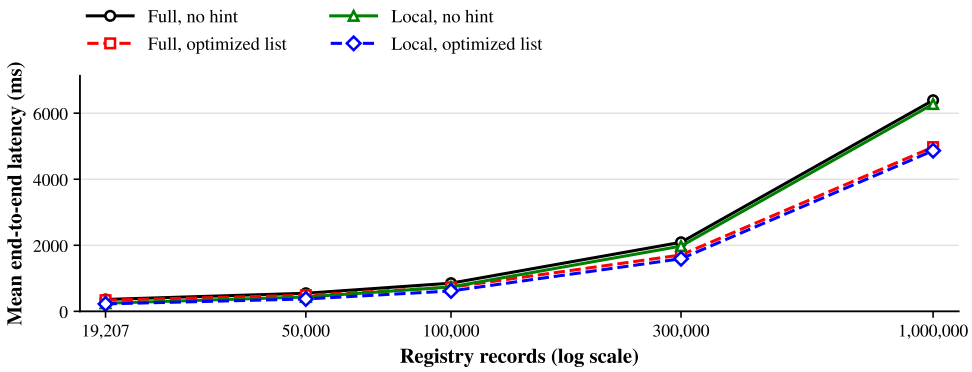


Figure 3: End-to-end verification across five registry sizes on 233 queries with $K = 10$. Means include all queries and weight the 60 sources equally. At one million records, each hint route visits 228/233 query targets.

Figure 3 shows the end-to-end speedup from hints within Local increasing from $1.088\times$ at 19,207 records to $1.240\times$ at 300,000 records. Both routes visit every query's target at each tested size through 300,000 records. Accepted sets and runtime states agree between routes within each predicate at these sizes.

At 300,000 records, hints reduce Local from 1.975 to 1.593 s with unchanged accepted sets. At one million, latency falls from 6.276 to 4.865 s with 228/233 targets visited. Five queries exhaust $K = 10$, changing four Local outputs from `SUPPORTED` to `INDETERMINATE`. Detailed results for all 4,660 live executions appear in Appendix G.

4.4 Isolating Sentence Selection

Table 1 crosses selection and aggregation using saved scores and frozen thresholds. At the same minimum threshold, selection matches all 233 decisions and reduces pairs from 2,994 to 1,292, or 56.85%. For 173 edited queries, the reduction is 76.63%. Unchanged queries retain all sentences.

Table 1: Factorial analysis on 233 fresh queries. S, C, and U are accepted counts for Supported, Contradicted, and Unknown. Agreement uses all/minimum as the reference. The two minimum rules share a threshold.

Rule	S	C	U	AUC S/C	AUC S/(C+U)	Agree	Pairs
All / minimum	130/132	0/51	15/50	0.9921	0.9575	233/233	2,994
All / mean	129/132	0/51	12/50	0.9972	0.9637	229/233	2,994
Selected / minimum	130/132	0/51	15/50	0.9921	0.9583	233/233	1,292
Selected / mean	130/132	0/51	14/50	0.9915	0.9641	232/233	1,292

Local combines 98.48% recall with 90.28% precision. It accepts 130/132 *Supported*, 0/51 *Contradicted*, and 14/50 *Unknown* queries. The false acceptance rate is 14/101, or 13.86%. Full and Local agree on 232/233 decisions. Matched random windows support selecting the observed change over an arbitrary window of equal length. A separately timed 39-pair benchmark reduces the sum of median semantic times from 25.609 to 4.010 s. Appendix B reports these controls and class intervals.

4.5 Assessment with Multiple Evidence Chunks

The 90-query long-document study directly evaluates the semantic component with multiple evidence chunks. Scoring pairs decrease from 51,039 to 1,672, or 96.72%, with AUC 0.9811 and 70.73% precision. Local accepts 29/30 *Supported*, 9/30 *Contradicted*, and 3/30 *Unknown* queries. Table 4 in Appendix B separates selection and aggregation across four rules. Section 5.3 uses paired chunks to explain how evidence aggregation changes support scores.

4.6 Retrieval Cost and Candidate Capacity

Table 2 quantifies net discovery savings with online hint decoding included. All six backend comparisons preserve 100% source recall. At 19,207 records, hints reduce mean discovery time by 23.49% for BM25 and 23.80% for hybrid retrieval. Dense retrieval illustrates the cost crossover: 8.04 ms without hints becomes 9.74 ms with hints, including 4.84 ms decoding. Section 5.2 explains these outcomes through avoided retrieval work.

Table 2: Discovery cost across retrieval backends on 233 queries with $K = 10$. Means include online query processing and hint decoding. Positive reduction denotes lower mean time. Every row reaches 100% source recall.

Records	Backend	No hint (ms)	Hint (ms)	Reduction
2,000	BM25	17.22	16.67	+3.24%
2,000	Dense	5.65	7.70	-36.23%
2,000	Hybrid	22.78	20.79	+8.71%
19,207	BM25	141.46	108.23	+23.49%
19,207	Dense	8.04	9.74	-21.16%
19,207	Hybrid	150.33	114.54	+23.80%

The budget sweep varies K from 3 to 50 and masks prefixes to quantify candidate demand. At 2,000 and 19,207 records, both routes visit each query's target at every tested budget with 20-bit prefixes. Table 6 in Appendix C shows that coarser prefixes need more candidate capacity. At 19,207 records, eight effective bits with $K = 10$ give 3.4% recall. Every miss under crowding coincides with hint candidates filling K before fallback. Random flips of one to three bits retain full recall in the tested sparse registries. Appendix C reports recall matrices and bucket occupancy.

4.7 Encoder Recovery, Work, and Quality

Figure 6 compares three strategies and caps on 24 prompts across Qwen3-8B, Mistral-7B, and OLMo-2-7B. At 48k, adaptation raises recovery over Extended-256 from 20 to 23 for Qwen. Mistral rises from 20 to 22, and OLMo from 23 to 24. The selected Qwen configuration recovers 40/60 fresh frames. The other 20 signed originals use plain fallback. Verification includes both paths. Completed hint attempts average 41,698 forward input tokens versus 421.6 for plain generation. Human review yields 42 ties and nine preferences per method. Table 13 and Appendix E.5 detail work and quality.

4.8 External Carrier Comparison

ORBIT-20 and the official RSBH implementation both recover 20-bit payloads on all 18 matched prompts (Qu et al., 2025). Median sampled tokens are 4,832 and 512, respectively. Table 14 reports recorded times under each implementation's timing scope, and Appendix E.6 gives paired quality judgments.

4.9 Authentication Boundary Checks

All 14 tests and 22 subchecks pass with real Ed25519 verification and SQLite. They exercise tampering, authentication before scoring on every route, and forged hints with bounded fallback. Appendix E.7 gives complete coverage.

5 Analysis and Discussion

5.1 Integrity of Authenticated Records

We establish the integrity of the object defined in Equation (2). The argument assumes an injective canonical encoding over allowed receipt cores and externally fixed trusted verification keys. Every accepting path enforces the same record authentication checks. The notation H_r and H_x distinguishes receipt and text inputs to the same SHA-256 function.

Proposition For a fixed trusted verification key, let $\mathsf{BadRecord}$ be the event defined in Section 3.3. For every polynomial time adversary with adaptive access to record issuance, there exist reductions R_s, R_r, R_x such that

$$\begin{aligned}\Pr[\mathsf{BadRecord}] \le\ & \mathrm{Adv}^{\text{EUF-CMA}}_{\mathsf{Sig}}(R_s) \\ & + \mathrm{Adv}^{\mathrm{CR}}_{H_r}(R_r) + \mathrm{Adv}^{\mathrm{CR}}_{H_x}(R_x).\end{aligned} \tag{5}$$

An unissued receipt core yields either a signature forgery or a collision in the receipt hash. Changed original bytes under an issued core yield a collision in the text hash. A union bound combines these cases for the two uses of SHA-256. Appendix F.1 gives the complete reduction, including locator consistency and legitimate replay.

Routing corollary Every accepted record retains this integrity property under arbitrary candidate proposals, including corrupted hints. The mandatory authentication step binds the accepted identifier to its issued core and original bytes. Routing determines which authenticated originals are examined and how the shared candidate budget is used. Source recall and semantic relation labels measure these aspects of the complete system. The real Ed25519 checks in Appendix E.7 exercise the common authentication step across discovery routes.

5.2 When Routing Hints Reduce Work

Let N be the registry size and $\mathcal{B}_u$ the target's bucket for a correct h-bit prefix. Modeling routing prefixes as independent uniform h-bit strings,

$$\mathbb{E}[|\mathcal{B}_u|] = 1 + \frac{N-1}{2^h}. \tag{6}$$

The target contributes one record and each other identifier shares its prefix with probability 2^{-h}. This explains how partial identifiers narrow discovery even when buckets contain multiple records.

Indexed lexical search also uses a candidate subset. Equation (6) describes occupancy, while the experiments measure runtime.

Let G_0 denote discovery cost without a hint and $R_h = D + P + FG_f$ the hinted discovery cost. Here D is decoding, P is bucket processing, $F \in \{0, 1\}$ indicates executed fallback, and G_f is its additional search cost. Subsequent verification costs V_0, V_h include authentication, eligibility, and semantic assessment. Then

$$\begin{aligned} C_0 = G_0 + V_0, \qquad C_h = R_h + V_h, \\ \mathbb{E}[C_0 - C_h] = \mathbb{E}[G_0 - R_h] + \mathbb{E}[V_0 - V_h]. \end{aligned} \tag{7}$$

With the assessment rule fixed, this decomposition isolates the net verification savings from hints. One issuance can serve repeated checks of redistributed text. Its embedding cost is amortized across these checks.

Measured routing savings Within Local, absolute savings rise from 19.88 ms at 19,207 records to 382.58 ms at 300,000 records. A separate CPU probe on 12 queries measures mean BM25 time increasing from 114.88 to 5,773.76 ms across the five sizes. Decoding stays near 4 ms. These measurements connect the larger routing gains to increasing retrieval cost. Table 2 reports the cost tradeoff across retrieval backends. The component measurements are reported in Appendix G.

Candidate capacity With 20-bit hints and $K = 10$, every query's target is visited at each tested size through 300,000 records. At one million, five queries spend all ten slots on non-target hint candidates before fallback. At 19,207 records, prefix masking reaches full recall with $K = 20$ at 14 bits or $K = 50$ at 12 bits. Deduplicated candidates from multiple decoded buckets contribute to this demand. These measurements connect registry size and prefix resolution to the shared candidate budget.

5.3 Efficiency and Scope of Local Assessment

Full requests mq scoring pairs and Local requests $|J_\rho|q$. The ratio $|J_\rho|/m$ is the retained fraction of query work. Each selected sentence remains paired with all source chunks under the same input rules. The fresh comparison at an identical minimum threshold isolates this saving while preserving all 233 decisions. Matched random windows and separately measured semantic runtime connect the gain to edit location and actual inference work.

Multiple evidence chunks Across 30 sources with 3 to 14 chunks, Local reduces scoring pairs by 96.72% and reaches AUC 0.9811. It retains 29/30 *Supported* queries, compared with 16/30 for Full with minimum aggregation. At their frozen thresholds, Local accepts 12/60 unsupported queries and Full accepts 8/60. Local's precision is 70.73%. Table 4 reports all four combinations, relating retained support and false acceptance to selection and aggregation.

Evidence aggregation For all 60 negative queries, the target sentence scores below the historical minimum threshold against the original claim's chunk. Adding other chunks moves 16 target scores across the threshold, comprising 12 *Contradicted* and four *Unknown* queries. In 58/60 cases, another chunk supplies the maximum. Document decisions also depend on other selected sentences and aggregation. This paired comparison isolates how additional evidence chunks change target sentence support scores. Selection determines which claims receive computation, while aggregation converts their evidence into a decision. Both use the same original evidence.

6 Conclusion

ORBIT reduces discovery and semantic work through compact hints and observed edits within one authenticated pipeline. The combined configuration reduces mean total latency by 37.44% at 19,207 records. Matched thresholds establish the computational value of sentence selection. Hints yield $1.240\times$ speedup within Local at 300,000 records with unchanged accepted sets. Local achieves 98.48% recall and 90.28% precision on fresh queries. The long-document component study reaches 96.67% recall and 70.73% precision. Paired evidence analysis distinguishes aggregation from selection. These results attribute system savings to complementary computational choices under the same authentication interface.

AI USE STATEMENT

Generative AI supplied reference annotations for human review and assisted manuscript editing and figure code. Final semantic and quality judgments were submitted through the human workflow in Appendix D and checked against archived texts, hashes, and receipts. Generative AI tools also assisted with experimental code generation, debugging, and the preparation of reproducibility scripts. AI-assisted audits examined implementation consistency and cross-checked reported results, figures, and tables against archived experimental outputs. The authors made the final methodological and editorial decisions and take responsibility for the code, analyses, artifacts, content, claims, and reported findings.

REFERENCES

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A human generated MAchine Reading COmprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016. URL https://arxiv.org/abs/1611.09268.

Maciej Besta, Lorenzo Paleari, Marcin Copik, Robert Gerstenberger, Ales Kubicek, Piotr Nyczyk, Patrick Iff, Eric Schreiber, Tanja Srindran, Tomasz Lehmann, Hubert Niewiadomski, and Torsten Hoefler. CheckEmbed: Effective verification of LLM solutions to open-ended tasks, June 2025. URL https://arxiv.org/abs/2406.02524.

C2PA. C2PA and content credentials explainer. C2PA Specifications, release 2.4, 2026a. URL https://spec.c2pa.org/specifications/specifications/2.4/explainer/Explainer.html.

C2PA. C2PA soft binding API. C2PA Specifications, release 2.4, 2026b. URL https://spec.c2pa.org/specifications/specifications/2.4/softbinding/Decoupled.html.

C2PA. Content credentials: C2PA technical specification. Technical specification, version 2.4, April 2026c. URL https://spec.c2pa.org/specifications/specifications/2.4/specs/C2PA_Specification.html.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In Shipra Agrawal and Aaron Roth (eds.), *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pp. 1125–1139. PMLR, 2024. URL https://proceedings.mlr.press/v247/christ24a.html.

Databricks. databricks-dolly-15k. Dataset card, version 1.0, 2023. URL https://huggingface.co/datasets/databricks/databricks-dolly-15k/blob/bdd27f4d94b9c1f951818a7da7fd7aeea5dbff1a/README.md. Revision bdd27f4d94b9c1f951818a7da7fd7aeea5dbff1a; CC BY-SA 3.0.

Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Mingyuan Wang. Publicly-detectable watermarking for language models. *IACR Communications in Cryptology*, 1(4), 2025. ISSN 3006-5496. doi: 10.62056/ahmpdkp10. URL https://cic.iacr.org/p/1/4/31.

Haocheng Fu, Yuqi Qian, Luyao Wang, and Yun Cao. DHMark: Public-key watermarking for LLM-generated text via Diffie-Hellman-guided rejection sampling, 2026. URL https://arxiv.org/abs/2608.03093v1. arXiv preprint, version 1.

Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. SemStamp: A semantic watermark with paraphrastic robustness for text generation. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4067–4082, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.226. URL https://aclanthology.org/2024.naacl-long.226/.

Simon Josefsson and Ilari Liusvaara. Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032, January 2017. URL https://www.rfc-editor.org/info/rfc8032/. Informational RFC.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/d78e9e4316e1714fbb0f20be66f8044c-Abstract-Conference.html.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 27469–27500. Curran Associates, Inc., 2023. doi: 10.52202/075280-1195. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/575c450013d0e99e4b0ecf82bd1afaa4-Abstract-Conference.html.

Kundan Krishna, Sanjana Ramprasad, Prakhar Gupta, Byron C. Wallace, Zachary C. Lipton, and Jeffrey P. Bigham. GenAudit: Fixing factual errors in language model outputs with evidence, 2024. URL https://arxiv.org/abs/2402.12566.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=FpaCL1MO2C.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177, 2022. doi: 10.1162/tacl_a_00453. URL https://aclanthology.org/2022.tacl-1.10/.

Gregory Kang Ruey Lau, Xinyuan Niu, Hieu Dao, Jiangwei Chen, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. Waterfall: Scalable framework for robust text watermarking and provenance for LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20432–20466, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1138. URL https://aclanthology.org/2024.emnlp-main.1138/.

He Li, Xiaojun Chen, Jingcheng He, Zhendong Zhao, Shuguang Yuan, Xin Zhao, and Yunfei Yang. ARGH-Mark: Anchor-synchronized watermarking with hamming correction for robust and quality-preserving LLM attribution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(44): 37583–37590, March 2026. doi: 10.1609/aaai.v40i44.41092. URL https://ojs.aaai.org/index.php/AAAI/article/view/41092.

Huijia Lin, Kameron Shahabi, and Min Jae Song. Unforgeable watermarks for language models via robust signatures. In Nadia Heninger and Mike Rosulek (eds.), *Advances in Cryptology – CRYPTO 2026*, volume 16806 of *Lecture Notes in Computer Science*, pp. 99–131, Cham, 2026. Springer. doi: 10.1007/978-3-032-35415-0_4. URL https://link.springer.com/chapter/10.1007/978-3-032-35415-0_4.

Mistral AI. Model card for Mistral-7B-Instruct-v0.3. Official model card, Hugging Face, May 2024. URL https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3. Model released May 22, 2024; version v0.3. Distinct from the 2023 Mistral 7B v0.1 report.

Wenjie Qu, Wengrui Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. Provably robust multi-bit watermarking for AI-generated text. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 201–220, Seattle, WA, August 2025. USENIX Association. ISBN 978-1-939133-52-6. URL https://www.usenix.org/conference/usenixsecurity25/presentation/qu-watermarking.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://www.nowpublishers.com/article/Details/INR-019.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. Can AI-generated text be reliably detected? stress testing AI text detectors under various attacks. *Transactions on Machine Learning Research*, January 2025. URL https://openreview.net/forum?id=OOgsAZdFOt.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 OLMo 2 furious, 2025. URL https://arxiv.org/abs/2501.00656. Technical report, version 3; first submitted December 31, 2024. A distinct shorter version appeared at COLM 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. AlignScore: Evaluating factual consistency with a unified alignment function. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11328–11348, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.634. URL https://aclanthology.org/2023.acl-long.634/.

Junlin Zhu, Baizhou Huang, and Xiaojun Wan. QuantileMark: A message-symmetric multi-bit watermark for LLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6790–6806, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.308. URL https://aclanthology.org/2026.acl-long.308/.

A Frozen Implementation Parameters

Carrier generation The primary frame contains a routing prefix of 20 bits and a CRC of four bits. Eight symbols carry three bits each. The legacy frame contains 23 prefix bits and nine symbols. Both profiles use an initial anchor of four tokens, 30 tokens per symbol, one repetition, and a maximum output length of 512 tokens. Candidate blocks use random streams indexed by counters and inverse CDF sampling. Batches contain 16 candidates and commit the earliest valid match. The final tokenization is checked before issuance. A plain fallback completes attempts that fail the frame, tokenization, or text checks. The issuer signs the digest of the final text on either path.

Base-128 allows 128 candidates per symbol. Extended-256 allows 256 candidates under the same sampling distribution. Adaptive-256 uses the extended limit and broadens the distribution when diversity is low. The trigger requires at least 32 unsuccessful candidates, at least 16 valid candidates, and a unique block fraction of at most 0.5. Ordinary sampling uses temperature 0.7, top-p 0.8, and top-k 20. Broadened sampling uses 0.9, 0.95, and 0, respectively. Adaptation resets for each symbol. Anchor, tail, and plain fallback generation use ordinary sampling.

The fresh cohort uses a cap of 48,000 forward input tokens. The work ledger includes failed batches, prompt and prefix replays, consistency checks, and fallback. The encoder reserves an allowance for plain fallback before further search. The text gate checks length and excessive repetition of four-grams. Generation uses bfloat16. Identifiers are preallocated deterministically from hashes with domain separation over the source identifier. Their 20 prefix bits are specified independently.

Discovery and eligibility List decoding tests the observed token sequence and sequences obtained by deleting one token before position $\min(n, L + 1)$. Here n is the query token count and L is the carrier span. The maximum is $L + 2$ attempts before prefix deduplication. This gives 246 attempts for the primary profile and 276 for the legacy profile. The optimized decoder prepares headers and token bytes once per query. It memoizes symbol hashes by symbol index and exact anchor and block bytes. The cache ends with the query, preserving candidate enumeration and checksum tests. The original fresh ablation uses the earlier list decoder.

SQLite FTS5 uses the unicode61 tokenizer. The query joins the first 128 distinct Unicode word terms with OR. BM25 ranks the results, with record identifiers breaking ties. Eligibility applies NFC normalization and case folding. It requires set Jaccard similarity of at least 0.8 over alphanumeric words and at least 32 tokens identified as words per text. Each input is capped at 4,096 lexical tokens and 100,000 characters. Declared language must be English.

Scoring and software Difference selection uses radius one over the worker's query segmentation. Every selected sentence is paired with the unchanged source chunks under the scorer's tokenization and truncation rules. The all/minimum and selected/minimum thresholds are both 0.3578843176364899. The all/mean threshold is 0.9377014071337487. The selected/mean threshold is 0.7778514524300895.

The original fresh ablation uses PyTorch 1.13.1+cu117, Transformers 4.30.2, and NLTK 3.8.1. The unified live and long-document studies use PyTorch 2.14.0+cu130 and Transformers 4.44.2, with NLTK 3.8.1 and 3.10.3 installed, respectively. Both later studies consume frozen sentence strings without runtime sentence tokenization. AlignScore uses upstream commit `a0936d5`, float32 inference, and TF32 disabled. The carrier tokenizer bridge uses Transformers 4.57.6. The original fresh ablation records an NVIDIA RTX 4090 and 49,140 MiB of device memory. The unified live study records an NVIDIA RTX 5090 and 32,607 MiB, with batch size 16. The long document study uses an NVIDIA GeForce RTX 4090 D and batch size 32. Both the unified and long document studies use float32 inference with TF32 disabled. The execution receipts retain full hashes for the model, profile, checkpoint, tokenizer, and source versions.

Threshold selection Calibration uses 104 weighted records from 21 sources in seven prompt clusters. The original AI labels contain 82 *Supported* and 22 *Contradicted* records. For each selection and aggregation rule, calibration maximizes accepted *Supported* records under an empirical *Contradicted* acceptance cap of 1%. Ties select the lower threshold. With 22 negative records, this criterion selects zero negative acceptances in calibration. Replay on the frozen sentence path reproduces all four thresholds exactly. Human review subsequently assigns 50 *Supported* and 27

Contradicted labels to the 77 distinct calibration pairs. The original thresholds remain fixed for the archived 39 pairs, 233 fresh queries, and 90 long document queries.

B Sentence Selection and Evidence Analysis

B.1 Fresh Cohort Metrics

Table 1 crosses sentence selection with score aggregation. All four rules reuse the same saved sentence and chunk scores. The two minimum rules share the same threshold, which isolates sentence selection. Their 233 decisions agree while scoring pairs decrease from 2,994 to 1,292. For the 173 edited queries, pairs decrease from 2,221 to 519, or 76.63%. The 60 unchanged queries use all sentences. The deployed Full and Local rules agree on 232 of 233 decisions.

Table 3: Local metrics on the fresh cohort under completed human review. Intervals are two-sided 95% Clopper–Pearson binomial reference intervals. Source-cluster bootstrap intervals are reported below.

Metric	Count	Rate	95% reference interval
Supported acceptance	130/132	98.48%	[94.63%, 99.82%]
Contradicted acceptance	0/51	0.00%	[0.00%, 6.98%]
Unknown acceptance	14/50	28.00%	[16.23%, 42.49%]
Precision among accepted queries	130/144	90.28%	[84.23%, 94.58%]
False acceptance rate	14/101	13.86%	[7.79%, 22.16%]

The false acceptance denominator contains 51 *Contradicted* and 50 *Unknown* queries. Table 3 reports binomial reference intervals for each metric. Source bootstrap intervals from 5,000 resamples are [16.33%, 39.62%] for *Unknown* acceptance and [86.09%, 94.41%] for precision. All 233 fresh queries use one evidence chunk. Sources contain 206 to 329 words, and all inputs fit within the scoring model's limit. Each edited query selects three sentences.

B.2 Controls for Selection and Runtime

An archived cohort contains 39 scored pairs from 20 sources. Its test subdivision contains 26 pairs from 13 sources. The S/C AUC comparison uses 15 *Supported* and 10 *Contradicted* pairs. The remaining test pair is *Unknown*. Twenty fixed seeds draw uniform contiguous windows with the same sentence count as the difference selector. This comparison holds the selected length fixed and changes the location rule.

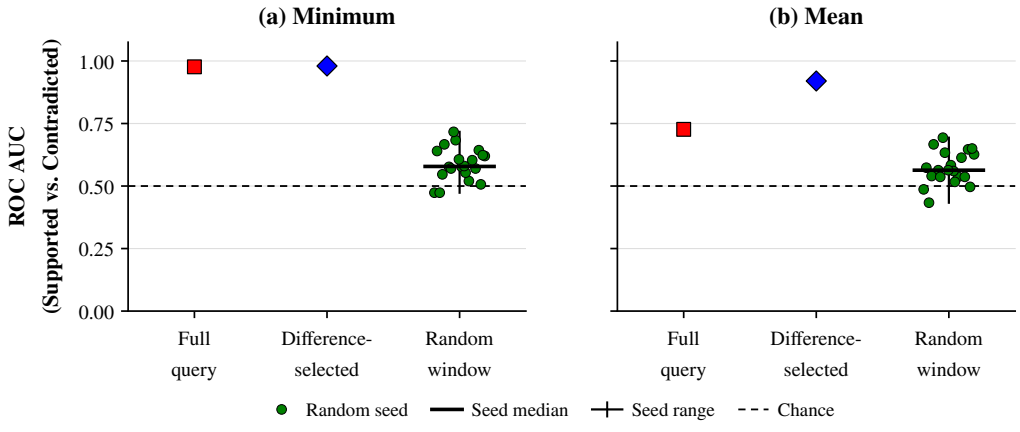


Figure 4: Selection controls on the archived test subdivision. Each random point is one fixed seed. Its window matches the number of sentences selected from the observed edits. S/C AUC uses 15 *Supported* and 10 *Contradicted* pairs.

At matched selection lengths, the observed edit provides an informative location for assessment (Figure 4). Random windows have median AUC 0.5783 with minimum aggregation and 0.5633 with mean aggregation. Their ranges are 0.4733 to 0.7167 and 0.4333 to 0.6933. At the frozen thresholds, all/minimum and all/mean accept 6 and 3 of the 15 *Supported* pairs. Both selected rules accept 13 of 15. The rules using all sentences accept 0 of 10 *Contradicted* pairs, and the selected rules each accept 1 of 10. These paired counts report retained support and false acceptance alongside the ranking comparison.

We time Full and selected sentence scoring separately on all 39 pairs in the archived cohort. Three timed repetitions alternate execution order. The sum of query median times decreases from 25.609 s to 4.010 s. Separate audit calls count 1,728 and 165 scoring pairs. All 39 input and score parity checks pass. The timer covers selection, live inference, and aggregation. Retrieval, authentication, and audit instrumentation are measured in their respective experiments.

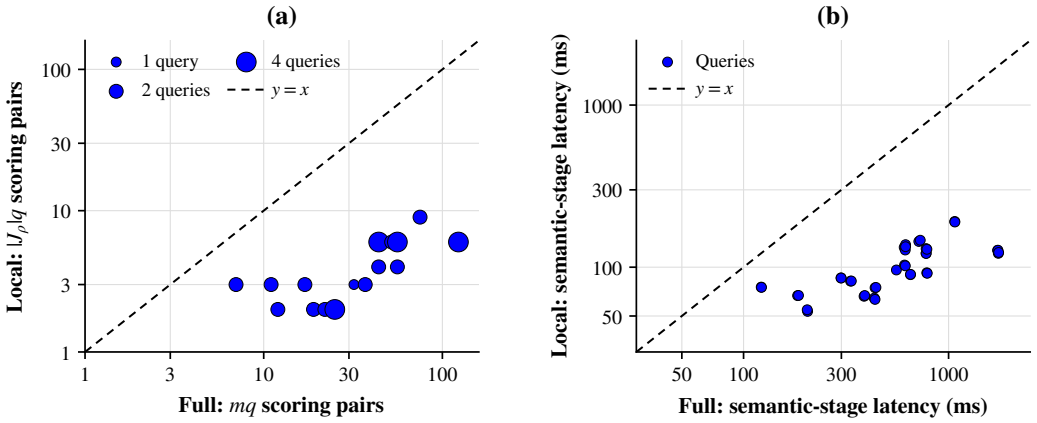


Figure 5: Paired work and runtime on 39 queries from 20 sources. The left panel reports scoring pairs from separate audit runs. The right panel reports median semantic runtime over three executions of each mode.

B.3 Long Documents and Evidence Chunks

Table 4: Scorer-level factorial analysis on 90 long-document queries. S, C, and U are accepted counts for Supported, Contradicted, and Unknown. Agreement uses all/minimum as the reference. Each rule uses its frozen threshold.

Rule	S	C	U	AUC S/(C+U)	Agree	Pairs
All / minimum	16/30	6/30	2/30	0.8017	90/90	51,039
All / mean	23/30	23/30	23/30	0.6950	45/90	51,039
Selected / minimum	29/30	11/30	3/30	0.9822	71/90	1,672
Selected / mean	29/30	9/30	3/30	0.9811	67/90	1,672

Table 4 reports 90 edits to 30 natural texts with 3 to 14 evidence chunks. Human review assigns 30 queries to each relation class. Selection reduces scoring pairs from 51,039 to 1,672, or 96.72%. For selected/mean, exact 95% intervals are [82.78%, 99.92%] for *Supported* acceptance and [14.73%, 49.40%] for *Contradicted* acceptance. They are [2.11%, 26.53%] for *Unknown* acceptance and [54.46%, 83.87%] for precision.

The paired chunk analysis holds the target sentence fixed to isolate the contribution of evidence aggregation. For each of the 60 negative queries, it first scores the sentence against the chunk containing the original claim. It then takes the maximum score over all source chunks. All 60 initial scores lie below the historical minimum threshold. Adding the remaining chunks moves 12 *Contradicted* and four *Unknown* target sentence scores above it. A different chunk supplies the

maximum in 58 of 60 cases. This comparison identifies which evidence drives the score change. Document acceptance additionally depends on the other selected sentences and the aggregation rule.

C Retrieval Backends and Candidate Budgets

C.1 Discovery Protocol

The supplement reuses the 233 fresh queries at registry sizes of 2,000 and 19,207. Each query is evaluated with and without hints under BM25, dense retrieval, and hybrid retrieval. The shared candidate budget is $K = 10$. Source recall records whether the issued construction source appears among the visited identifiers. The source identifier is used to evaluate this discovery experiment. Its fallback starts when the hint stage has not visited that source. This isolates retrieval work before semantic assessment.

The two protocols connect source discovery to the complete verifier's acceptance rule. The hint stage finds the target source for 62 queries, leaving 171 for baseline retrieval in the discovery experiment. For 14 of these 62 queries, the complete verifier does not accept the target source, bringing the total number of BM25 calls to 185. Both protocols use the same texts and fixed candidate budget. These counts identify the semantic decision between finding a source and accepting its support.

Dense retrieval uses `all-MiniLM-L6-v2` embeddings of dimension 384. Normalized embeddings are searched with an exact cosine matrix product on CUDA. Hybrid retrieval applies reciprocal rank fusion with constant 60 and a pool of 100 from each backend. Online query encoding and fusion are included in discovery time. Index construction is measured separately. The dense index takes 21.86 s and 3,136,000 bytes for 2,000 records. For 19,207 records it takes 25.35 s and 30,116,576 bytes. The hybrid backend shares this dense index.

Timing and summaries Decoding is executed once per query and its measured cost is charged to every ordinary hint configuration. The reported total equals the measured routing time plus that external decoding time. The two registry runs have mean decoding costs of 4.05 and 4.84 ms. Table 5 reports arithmetic means, medians, and p95 values over the 233 queries. These describe average and distributional behavior within each run. The execution order evaluates the configuration without hints before the hint configuration for each backend and query.

Table 5: Discovery time distributions in milliseconds. Each value is computed over 233 queries. Ordinary hint times include decoding.

Records	Backend	Route	Mean	Median	p95
2,000	BM25	No hint	17.22	17.07	18.68
2,000	BM25	Hint	16.67	20.79	22.67
2,000	Dense	No hint	5.65	5.65	6.44
2,000	Dense	Hint	7.70	8.66	9.53
2,000	Hybrid	No hint	22.78	22.62	24.49
2,000	Hybrid	Hint	20.79	26.34	28.27
19,207	BM25	No hint	141.46	141.40	150.33
19,207	BM25	Hint	108.23	142.81	154.35
19,207	Dense	No hint	8.04	7.55	10.88
19,207	Dense	Hint	9.74	10.87	13.31
19,207	Hybrid	No hint	150.33	150.46	160.26
19,207	Hybrid	Hint	114.54	151.21	164.47

The backend comparison measures the balance between retrieval work avoided and online decoding cost. At 19,207 records, BM25 and hybrid yield mean net savings of 33.22 and 35.78 ms. Dense routing uses 4.89 ms internally, compared with 8.04 ms for its baseline. Adding 4.84 ms of decoding gives 9.74 ms total. The measured components therefore account for the net result in each backend. All six ordinary comparisons visit the source for every query.

C.2 Prefix Resolution and Budget

The budget sweep uses $K \in \{3, 5, 10, 20, 50\}$. It retains the original 20 prefix bits or masks decoded prefixes to 14, 12, 10, or eight effective bits. Masking changes bucket resolution within the same registry. It measures candidate pressure independently of adding records. Table 7 reports occupied bucket sizes. Their means average occupied buckets, whereas Equation (6) concerns the bucket of a fixed target.

Table 6: Source recall (%) at 19,207 records as effective prefix width and budget vary. Masking coarsens the decoded prefixes within the same registry. The native profile uses 20 bits.

Prefix bits	$K = 3$	$K = 10$	$K = 20$	$K = 50$
20	100.0	100.0	100.0	100.0
14	45.9	97.4	100.0	100.0
12	15.5	50.2	87.1	100.0
10	3.9	12.0	24.0	68.7
8	2.1	3.4	8.2	22.3

Table 7: Bucket occupancy at each effective prefix width. Mean and maximum size refer to occupied buckets.

Records	Prefix bits	Occupied buckets	Mean size	Maximum
2,000	20	1,999	1.001	2
2,000	14	1,878	1.065	3
2,000	12	1,554	1.287	4
2,000	10	869	2.301	8
2,000	8	255	7.843	17
19,207	20	19,037	1.009	2
19,207	14	11,328	1.696	7
19,207	12	4,064	4.726	15
19,207	10	1,024	18.757	33
19,207	8	256	75.027	95

Table 8: Complete source recall matrix (%) for decoded prefixes with masking. No hint, the oracle, and random errors of one to three bits reach 100% at every tested budget.

Records	Bits	$K = 3$	$K = 5$	$K = 10$	$K = 20$	$K = 50$
2,000	20	100.0	100.0	100.0	100.0	100.0
2,000	14	98.7	100.0	100.0	100.0	100.0
2,000	12	90.1	98.7	100.0	100.0	100.0
2,000	10	36.5	62.7	93.1	100.0	100.0
2,000	8	7.3	16.3	37.8	76.4	100.0
19,207	20	100.0	100.0	100.0	100.0	100.0
19,207	14	45.9	73.0	97.4	100.0	100.0
19,207	12	15.5	26.6	50.2	87.1	100.0
19,207	10	3.9	7.7	12.0	24.0	68.7
19,207	8	2.1	3.0	3.4	8.2	22.3

At 2,000 and 19,207 records with 20-bit prefixes, both routes visit the target source for all 233 queries at every tested budget. With 19,207 records and 14 effective bits, recall reaches 100% at $K = 20$. With 12 effective bits, it reaches 100% at $K = 50$. These results quantify how additional candidate capacity compensates for coarser prefixes.

The collision records locate source misses at the shared candidate budget. Every missed source coincides with hint candidates filling that budget before fallback can execute. At eight effective bits and $K = 10$, this accounts for all 225 misses at 19,207 records. The decoder returns a mean of 4.67

prefixes before masking. Their deduplicated candidate sets jointly determine the demand on K. This links prefix resolution, bucket occupancy, and the capacity available for fallback.

C.3 Random Prefix Errors

The corruption conditions flip one, two, or three bits of the true prefix. They sample errors with seed 20260923. Each budget configuration samples its own corrupted prefixes. Each condition supplies its prefix directly and measures candidate visits from that point. At both registry sizes, every tested corruption condition and budget reaches 100% source recall. Averaged over the 15 corruption configurations, hint candidates consume 0.0034 slots per query at 2,000 records and 0.0155 at 19,207. The sparse index leaves the remaining budget available for fallback.

The oracle condition supplies the true prefix directly. Collision and corruption timings likewise begin from their supplied prefixes. Their primary measurements are recall and candidate use. The ordinary hint configurations include text decoding and supply the online cost comparison. The archive retains all 26,562 records across the two retrieval runs and two budget runs. The table generation script checks shared budgets, recall, and external decoding charges against these records.

D Data and Review Protocol

The source package contains the aggregate data and figure scripts used in the paper. It also includes the routing supplement records and a script that regenerates their tables. Each experiment retains its own cohort, timing scope, and statistical unit. Model execution uses the weights and environments recorded with the corresponding study. Signing keys and account credentials remain with their owners.

D.1 Data Construction

Fresh queries The prompts come from databricks-dolly-15k version 1.0 at revision `bdd27f4` (Databricks, 2023). Selection uses seed `orbit-final60-20260918-v1`. It draws 30 general QA and 30 brainstorming prompts with empty reference contexts. Generation and editing use the same Qwen3-8B checkpoint and tokenizer. Inference uses bfloat16, temperature 0.7, top-p 0.8, top-k 20, and thinking disabled.

The editor segments the original with trained English Punkt. It selects a declarative sentence containing 12 to 70 words separated by whitespace. The prompt supplies the full original, the selected sentence, and one of three instructions. These request preservation of all claims, explicit contradiction of one substantive claim, or addition of a plausible detail neither supported nor contradicted by the original. The editor returns one replacement sentence in JSON with a single draw capped at 160 tokens. Bytes outside the selected span remain fixed. The 240 planned slots yield 233 valid queries.

Long documents The semantic-component study contains 30 natural texts from the same Dolly dataset, spanning 1,001–4,274 words and 3–14 evidence chunks. Each source contributes three single-sentence edits at a common factual anchor within its first 11 sentences. The edited fact lies in the first evidence chunk for 87 queries and the second for three. Final human review assigns one *Supported*, one *Contradicted*, and one *Unknown* query per source. All 90 pairs receive direct semantic scoring. Nine pairs from three sources exceed the deployed limit of 4,096 lexical tokens. Reported acceptances are threshold decisions from the scoring component. Texts, thresholds, and analysis rules were frozen before model scoring and the subsequent human review.

D.2 Human Review

The completed review covers 572 items. These comprise 439 semantic relation items and 133 output quality items. Semantic review covers 233 fresh queries, 77 calibration pairs, 39 mechanism pairs, and 90 long document queries. Quality review covers 60 fresh pairs, 54 carrier pairs, and 19 deduplicated encoder strategy pairs. Identity fields, text hashes, item identifiers, and completion status are validated for every item.

The review records provide one final human judgment per item. The interface displays archived AI reference judgments, and final submissions use role `AUTHOR`. Two recorded identifiers track submission streams that partition the items into disjoint sets. Reviewer headcount and agreement between reviewers are unmeasured in this review protocol. The completed labels supply 132 *Supported*, 51 *Contradicted*, and 50 *Unknown* fresh queries. The long document labels contain 30 queries in each class. These item labels determine the reported semantic metrics.

A query is *Supported* when the original states its substantive claims or supports them through valid textual inference. It is *Contradicted* when the original supports an incompatible claim. It is *Unknown* when the original supplies insufficient support. An internally inconsistent original receives an unable to judge label. Scores, thresholds, and runtime decisions remain frozen throughout this review. Human judgments determine the final class metrics and quality summaries.

The study evaluates generated texts and archived verification records. Deployment access controls protect retained originals. Support decisions refer to the authenticated record and the configured semantic policy.

D.3 Cohorts and Statistical Units

Table 9: Experimental cohorts and execution units. Repeated configurations retain their underlying source cohort.

Experiment	Sources / queries	Configurations and observations	Measurement
Unified live	60 / 233	3 sizes $\times$ 2 routes $\times$ 2 predicates. 2,796 primary and 1,920 repeat calls.	Complete verification with live inference.
Scale extension	60 / 233	5 sizes $\times$ 2 routes $\times$ 2 predicates. 4,660 primary calls.	Complete live verification with MS MARCO background passages.
Fresh ablation	60 / 233	2 predicates $\times$ 3 routes. 1,398 calls at 270 records.	Original instrumented live pipeline.
Retrieval supplement	60 / 233	2 sizes $\times$ 3 backends $\times$ 2 routes, plus oracle. 3,262 rows.	Discovery including ordinary hint decoding.
Budget supplement	60 / 233	2 sizes $\times$ 5 budgets $\times$ 10 conditions. 23,300 rows.	Source recall and candidate pressure.
Archived controls	20 / 39	20 matched random window seeds. 234 clean timings.	Selection and semantic runtime.
Long documents	30 / 90	Four semantic rules. 3 to 14 evidence chunks.	Live score matrix and aggregation analysis.
Encoder	24 prompts	3 models $\times$ 3 strategies $\times$ 3 caps. 648 outputs.	Recovery and actual input work.
Legacy carriers	18 prompts	Plain, ORBIT-20, ORBIT-23, RSBH. 72 outputs.	Payload and generation costs.
CPU score replay	18 / 90	3 sizes $\times$ 4 routes $\times$ 3 repetitions. 3,240 calls.	CPU discovery and authentication with archived scores.

Fresh, original unified, and scale extension means average queries within each source and then weight all 60 sources equally. Their paired bootstrap intervals use 5,000 source resamples. Repeated live executions are analyzed separately from primary executions. The routing supplement reports arithmetic means over its 233 queries. Its repeated configurations describe the same 60 sources.

E System Execution and Issuance Results

E.1 Unified Live Execution

The unified live study evaluates 233 queries at 270, 2,000, and 19,207 records. Full and Local each run with and without optimized list decoding. The study completes 2,796 primary executions and 1,920 additional executions on 80 fixed queries. All 43,236 semantic forward calls use live inference and complete successfully. Primary means weight the 60 sources equally.

Table 10: Unified live latency on 233 queries. Cells report the mean with equal source weights and the query p95, in milliseconds.

Records	Full No hint	Full Optimized list	Local No hint	Local Optimized list
270	246.82 / 311.73	251.03 / 318.39	129.11 / 273.81	132.53 / 273.43
2,000	256.26 / 322.71	258.59 / 323.73	137.48 / 284.92	139.86 / 276.48
19,207	363.08 / 433.13	343.86 / 436.27	244.63 / 391.44	224.32 / 354.25

Every original unified configuration visits the target source for all 233 queries within the shared budget $K = 10$. Optimized list decoding invokes BM25 for 185 queries and avoids it for 48. For a fixed predicate, accepted identifier sets and runtime states match across routes and registry sizes. The 80 repeated queries preserve these outputs across three measurements in all 12 configurations. The timer covers decoding, retrieval, record access, authentication, eligibility, selection, and synchronized semantic inference. Startup, warmup, and log output precede or follow the measured call.

E.2 Original Pipeline Ablation

The original ablation uses 60 fresh Dolly prompts, split equally between general QA and brainstorming. Qwen3-8B generates a plain response and a response with the frozen hint configuration for each prompt. The registry contains the 60 retained originals from the hint configuration and 210 existing records. Each retained original contributes its unchanged text and three attempted sentence edits, giving 233 valid queries from 240 planned slots. It crosses Full and Local with no hint, single decoding, and the original list decoder. Every configuration makes 233 semantic requests, one per query. Table 11 includes token audit instrumentation and worker communication in its time measurements.

Table 11: Original instrumented ablation on 233 queries and 270 records. Means weight sources equally. Candidate visits and scoring pairs are query totals. Every configuration makes one semantic request per query. The timer includes live inference, token auditing, and the original decoder.

Predicate	Discovery route	Latency (ms)	Candidate visits	Scoring pairs	Accepted	Indeterminate
Full	No hint	276.56	2,330	2,994	145	88
Full	Single decode	277.12	1,970	2,994	145	88
Full	List decode	347.73	1,898	2,994	145	88
Local	No hint	145.73	2,330	1,292	144	89
Local	Single decode	146.35	1,970	1,292	144	89
Local	List decode	217.21	1,898	1,292	144	89

Local reduces the original instrumented mean from 276.56 to 145.73 ms without hints. Single and list decoding reduce total candidate visits from 2,330 to 1,970 and 1,898. Within each predicate, route changes preserve accepted identifier sets.

E.3 Semantic Outcomes

Table 12: Human relations and runtime acceptance on 233 fresh queries. Accepted identifier sets match across the three original routes for each predicate.

Human relation	Queries	Full accepts	Local accepts
Supported	132	130	130
Contradicted	51	0	0
Unknown	50	15	14
Total valid	233	145	144

Table 12 reports all 233 queries under the completed human review. The labels comprise 132 *Supported*, 51 *Contradicted*, and 50 *Unknown* queries. Full and Local each accept 130 *Supported* queries and zero *Contradicted* queries. They accept 15 and 14 *Unknown* queries, respectively, and agree on 232/233 decisions. Appendix B gives rates and confidence intervals.

E.4 CPU Score Replay

The earlier replay study uses 90 queries from 18 sources and the legacy frame. It evaluates nested registries with 210, 2,000, and 19,207 records. Four routes and three repetitions produce 3,240 observations. Discovery and authentication execute on the CPU, while semantic scores are retrieved from the archive. Mean times without hints are 6.71, 15.86, and 91.92 ms. Optimized list decoding gives 5.57, 9.51, and 42.82 ms. At 19,207 records, the ratio is $2.15\times$ and the paired saving is 49.10 ms. The source bootstrap interval is $[41.39, 55.58]$ ms. Mean candidate visits are 10, 6, and 4.9 for no hint, single decoding, and optimized list decoding. Returned identifiers and semantic pair sets match across routes and repetitions.

E.5 Bounded Hint Generation

The development study uses 24 shared prompts for Qwen3-8B, Mistral-7B-Instruct-v0.3 (Mistral AI, 2024), and OLMo-2-1124-7B-Instruct (Team OLMo et al., 2025). Three strategies and three work caps produce 216 outputs per model and 648 overall. Base-128, Extended-256, and Adaptive-256 are named for their per symbol candidate limits. Adaptive-256 also broadens sampling when repeated unsuccessful blocks indicate low candidate diversity. All strategies share the frame format and maximum output length. The fresh cohort evaluates the chosen configuration on new prompts. The caps are 32,000, 48,000, and 64,000 forward input tokens per completed attempt. The ledger counts prompt prefills, replayed prefixes, discarded candidates, checks, and fallback. It measures processed input tokens. Sampled tokens, output length, FLOPs, and elapsed time are separate quantities.

Table 13: Encoder recovery and mean input work on 24 shared development prompts. Work is in thousands of forward input tokens (k). Each cap bounds a completed attempt. Outputs are separately capped at 512 tokens. Each model uses its own tokenizer.

		Base-128		Extended-256		Adaptive-256	
Model	Cap	Recovered	Input (k)	Recovered	Input (k)	Recovered	Input (k)
Qwen3-8B	32k	13/24	25.43	13/24	26.95	13/24	28.81
Qwen3-8B	48k	19/24	26.93	20/24	29.85	23/24	31.79
Qwen3-8B	64k	19/24	26.93	21/24	30.22	24/24	31.98
Mistral-7B	32k	15/24	25.52	15/24	25.97	15/24	26.22
Mistral-7B	48k	20/24	27.95	20/24	29.07	22/24	29.54
Mistral-7B	64k	20/24	28.51	20/24	29.85	22/24	30.10
OLMo-2-7B	32k	16/24	26.22	16/24	27.17	16/24	27.14
OLMo-2-7B	48k	22/24	27.83	23/24	29.31	24/24	29.66
OLMo-2-7B	64k	22/24	27.83	23/24	29.31	24/24	29.66

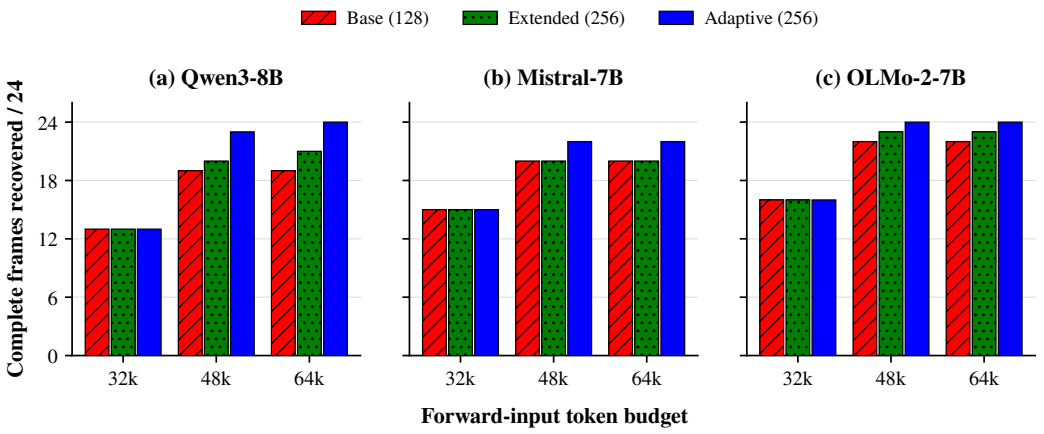


Figure 6: Complete frame recovery on the same 24 development prompts for each model. Strategy labels specify candidate limits per symbol. Table 13 gives actual work within each cap. The OLMo outputs at 48k and 64k are identical.

At the 48k cap, Adaptive-256 raises recovery over Extended-256 from 20 to 23 of 24 for Qwen. Mistral rises from 20 to 22 and OLMo from 23 to 24. The corresponding input work increases are 6.51%, 1.61%, and 1.22%. These paired measurements quantify the recovery gained together with the work used by adaptation. All 72 matched OLMo outputs at 48k and 64k have identical text, tokens, recovery, and work. For these prompts, the 48k budget already accommodates the measured OLMo outcomes.

The fresh cohort carries both issuance paths through the complete verification study. The selected Qwen Adaptive-256/48k configuration recovers 40 of 60 frames on new prompts, compared with 23 of 24 on development prompts. Plain fallback supplies the other 20 signed originals. All 60 originals enter the registry and verification experiments. Mean forward input work is 41,698 tokens for completed hint attempts and 421.6 for plain generation. These measurements link the observed hint availability and issuance cost to the evaluated system inputs.

Output quality The fresh comparison records paired preferences together with absolute content ratings. Among 60 hint/plain pairs, human review gives 42 ties and nine preferences for each method. Both methods receive major error flags on the same seven pairs. Mean task, coherence, and completeness scores are 3.333, 3.767, and 3.633 for hint outputs. Plain outputs score 3.200, 3.900, and 3.667. For 19 deduplicated encoder strategy pairs, baseline/adaptive/tie preferences are 6/1/12. These judgments complement recovery and work by reporting how the resulting texts are assessed.

E.6 External Carrier Comparison

Table 14: Carrier comparison on 18 shared prompts in six clusters. Sampled tokens, forward input tokens, and time are medians. Net/frame bits distinguish payload from coding overhead. All arms have a median output length of 512 tokens. Times use the scopes described below.

Method	Net/frame bits	Carrier tokens	Recovered	Sampled tokens	Forward input tokens	Recorded time (s)
Plain	0/0	—	—	512	536	—
ORBIT-20	20/24	244	18/18	4,832	27,542	33.87
ORBIT-23	23/27	274	17/18	5,522	33,150	—
Qu RSBH	20/30	—	18/18	512	536	1,408.05

The comparison places ORBIT's compact routing payload alongside an established message carrier on 18 prompts in six clusters (Qu et al., 2025). ORBIT-20 and the official RSBH implementation each recover 20 net bits on all 18 outputs. Table 14 reports their generation resources in the common local

environment. Median sampled tokens are 4,832 for ORBIT-20 and 512 for RSBH. Recorded median times are 33.87 s and 1,408.05 s, respectively. ORBIT's timer covers generation, frame checking, the text gate, and fallback. RSBH's timer covers generation and official payload decoding. Payload recovery and token accounting give the matched comparison, with elapsed times described under these implementation scopes.

Paired preferences and absolute flags describe complementary aspects of carrier output quality. For ORBIT-20/RSBH, human review gives 16 ties and one preference per method. For ORBIT-23/RSBH, it gives 14 ties, one ORBIT preference, and three RSBH preferences. For Plain/RSBH, it gives 15 ties, two Plain preferences, zero RSBH preferences, and one unable to judge item. Across the 54 pairs, 45 contain at least one major error flag. Reporting both measures separates relative carrier preference from the absolute quality of the generated content.

E.7 Authentication Checks

A separate CPU regression suite uses real Ed25519 verification and the SQLite record store. All 14 tests and 22 named subchecks pass. They cover altered source bytes, receipt fields, signatures, untrusted keys, duplicate JSON keys, and locator mismatches. They also exercise authentication before scoring on each route and forged hints under bounded fallback. Controlled decoder outputs and a fixed accepting scoring spy isolate the authentication and control flow behavior.

F Security Proofs

F.1 Proof of Integrity for Authenticated Records

Maintain a transcript of issued receipt cores, signed messages, and original bytes. Suppose an accepted record contains an unissued core $c^\star$. Its valid signature authenticates $m^\star = H_r(d \| \operatorname{enc}(c^\star))$. If $m^\star$ was never submitted for signing, the signature is an EUF-CMA forgery. Otherwise an issued core c has the same signed message. Since $c^\star \neq c$ and the encoding is injective, their distinct encoded inputs give a collision in H_r.

Now suppose the receipt core was issued, but the accepted original bytes differ from those recorded at issuance. Successful digest verification gives two distinct byte strings with the same H_x value. This is a collision in the text hash. The cases cover BadRecord. A union bound establishes Equation (5) for both hash uses, including when they use the same hash function. The locator check ensures that each accepted record has the identifier that was requested.

A legitimate replay preserves the issued core and committed bytes. It therefore satisfies the record integrity property. Ordinary EUF-CMA suffices because a second signature on an already signed message leaves that authenticated object unchanged. For multiple externally fixed issuer keys, summing the corresponding bounds gives the overall bound.

G Registry Scale Extension

Registry construction The registry scaling study preserves the signed bytes of the 270 base records. It adds passages from the MS MARCO Passage Ranking v1 collection (Bajaj et al., 2016). The five nested registries contain 19,207, 50,000, 100,000, 300,000, and 1,000,000 records. All sizes use one fixed ordering selected independently of the queries. Deduplication applies Unicode NFC, case folding, and whitespace normalization. The stored passage text remains unchanged. Normalized duplicates of the preserved records are excluded. A domain separated hash with seed 20260924 orders the background pool. Another domain separated hash assigns identifiers without forcing their routing prefixes. Background records have signed receipts with hint status `ABSENT`. The largest registry contains 999,730 added passages.

The 19,207-record registry is rebuilt from this background pool. Its background corpus differs from that used in Appendix E.1. All five registry sizes use the same 233 queries from 60 sources and their existing labels. It uses the archived 20-bit hint profile, the shared budget $K = 10$, and the frozen Full and Local thresholds. No new query, human label, or hint generation is introduced.

Execution and statistics The matrix contains 4,660 primary executions across five sizes, two routes, and two predicates. Query order is determined by a fixed hash ordering. The 20 configuration cells are cyclically rotated within each query. This batch contains one execution per query and configuration, with no additional repetition cohort. Eligible candidates use actual AlignScore inference without a semantic score cache. The completed run records 4,650 semantic requests and 42,773 forward calls. Ten executions have no semantic request because five queries exhaust the hint budget under each predicate at one million records.

The timer includes query tokenization, decoding, retrieval, record access, Ed25519 authentication, eligibility, sentence preparation, selection, and synchronized semantic inference. Startup, preflight checks, warmup, and log output occur outside the measured call. The runtime records an NVIDIA GeForce RTX 4090, four CPU threads, and batch size 16. The scoring environment records PyTorch `2.14.0+cu130`, Transformers `4.44.2`, and NLTK `3.8.1`. Inference uses float32 with TF32 disabled. The verifier prepares eligible candidates with the archived English Punkt model at runtime. The scorer consumes the resulting sentence strings.

Means first average queries within each source and then weight all 60 sources equally. All scheduled queries enter the reported means, including those with target misses. Paired intervals use 5,000 bootstrap draws over sources. The query p95 uses the recorded nearest-rank convention. Tables 15 and 16 report the complete results. The construction manifests and execution receipts accompany the result archive.

Table 15: Scale extension with $K = 10$ and 20-bit routing prefixes. Cells report the mean with equal source weights and query p95 in milliseconds. All 233 queries enter every cell, including those with target misses. At one million records, hint routes have five target misses and four changes in accepted sets per predicate.

Records	Full, no hint	Full, hint	Local, no hint	Local, hint
19,207	359.77 / 429.07	337.49 / 425.10	244.93 / 381.75	225.06 / 346.59
50,000	547.18 / 624.85	486.71 / 623.17	433.97 / 573.92	374.24 / 538.22
100,000	852.58 / 941.79	728.33 / 940.29	739.15 / 901.08	616.48 / 844.66
300,000	2090.65 / 2280.89	1706.56 / 2273.58	1975.19 / 2181.33	1592.61 / 2110.72
1,000,000	6391.19 / 6931.13	4972.97 / 6887.30	6275.62 / 6823.60	4865.23 / 6754.52

Table 16: Hint effects within each fixed predicate in the scale extension. Intervals use 5,000 paired source bootstrap draws. Visits give no hint / hint target counts out of 233 queries. Differences count queries with unequal returned identifier sets.

Records	Predicate	Speedup [95% interval]	Visits	Differences
19,207	Full	1.066× [1.050, 1.083]	233 / 233	0
19,207	Local	1.088× [1.063, 1.115]	233 / 233	0
50,000	Full	1.124× [1.094, 1.157]	233 / 233	0
50,000	Local	1.160× [1.119, 1.204]	233 / 233	0
100,000	Full	1.171× [1.130, 1.217]	233 / 233	0
100,000	Local	1.199× [1.150, 1.255]	233 / 233	0
300,000	Full	1.225× [1.171, 1.288]	233 / 233	0
300,000	Local	1.240× [1.181, 1.309]	233 / 233	0
1,000,000	Full	1.285× [1.220, 1.360]	233 / 228	4
1,000,000	Local	1.290× [1.223, 1.366]	233 / 228	4

Outputs and candidate capacity At every tested size through 300,000 records, both routes visit each query's target. Accepted identifier sets and runtime states agree between routes within each predicate. At 300,000 records, hints reduce Local's mean latency from 1.975 to 1.593 s. Its speedup is $1.240\times$, with a 95% source bootstrap interval of $[1.181, 1.309]$.

At one million records, hinted Local takes 4.865 s compared with 6.276 s without hints. The speedup is $1.290\times$, with interval $[1.223, 1.366]$. Both hinted predicates visit 228/233 query targets. For each predicate, the same five queries visit ten non-target hint candidates and exhaust the shared budget before BM25 fallback. Four queries change from `SUPPORTED` to `INDETERMINATE` under each predi-

cate. The fifth is already `INDETERMINATE` without hints. Thus the largest configuration records both latency savings and the effect of fixed candidate capacity on returned results. Every returned identifier in the 4,660 executions is the corresponding query's construction target. This observation concerns source identity, while the human relations continue to describe semantic support. The archive retains the individual affected queries in `analysis_results/parity_disagreements.jsonl`.

CPU component probe The component probe times 12 matched queries from three source groups. It measures decoding, bucket processing, BM25, and authentication separately from the live verification matrix. BM25 mean time grows from 114.88 to 5,773.76 ms across the five sizes. Decoder time remains near 4 ms. An untimed occupancy audit covers all 233 queries at each size. At one million records, five query candidate unions exceed $K = 10$. These counts connect the observed misses to the shared budget. Table 17 distinguishes the timing subset from the full occupancy audit.

Table 17: Separate CPU component probe on 12 matched queries from three sources. Times are query means in milliseconds. The final column counts queries whose full decoded candidate union exceeds $K = 10$ in the untimed audit of all 233 queries.

Records	BM25	Decode	Bucket	Union above K
19,207	114.88	4.40	0.121	0
50,000	292.02	4.07	0.089	0
100,000	577.35	3.87	0.077	0
300,000	1745.13	3.97	0.079	0
1,000,000	5773.76	4.01	0.089	5

H Relationship to C2PA

Provenance and recovery C2PA standardizes signed manifests, content bindings, and validation under a trust model (C2PA, 2026c). Its architecture supports external manifest storage and recovery through watermarks or fingerprints (C2PA, 2026b). Version 2.4 also specifies manifest embedding for unstructured and structured text (C2PA, 2026c). These mechanisms provide provenance infrastructure for applications.

Verification objects C2PA validation concerns provenance assertions and their association with an asset (C2PA, 2026a). ORBIT authenticates a retained original x and evaluates its support for an observed query y. A valid record for x can coexist with an edited y that contradicts x. The cryptographic commitment remains to x. The predicate $\Phi(x, y)$ evaluates the separate support relation. C2PA allows custom assertions, so an application could record such an assessment alongside provenance (C2PA, 2026a).

Computational policy The C2PA Soft Binding API includes a `maxResults` parameter that limits returned matches (C2PA, 2026b). ORBIT evaluates a cumulative budget of distinct visited identifiers across decoded prefix buckets and retrieval fallback. Unavailable records and failed authentication checks also consume this budget. Observed edits select query sentences, each scored against the retained source evidence under the fixed model input rules. The experiments measure the cost and outcomes of this policy. Their runtime comparisons hold ORBIT's authentication procedure fixed.

Potential integration A C2PA integration would validate a candidate original's manifest, trust chain, and content binding before supplying the original to semantic assessment. A usable manifest reference would permit direct lookup. Otherwise, recovery would supply candidate manifests for validation. This integration would require adapting the record format and trust handling. The evaluated implementation uses ORBIT's Ed25519 receipts and externally trusted keys. The comparison above specifies verification roles and integration requirements.