# PLAIT: FROM HALLUCINATION SCORES TO PARENT-PRESERVING AUDIT PLANS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Hallucination detectors estimate what may be wrong; human auditing must decide what is worth checking. Under a fixed review budget, these decisions are coupled: once a response and its evidence are open, additional claims become cheaper to inspect. We introduce PLAIT, which converts detector outputs into audit plans that account for these shared costs. A frozen detector, the *parent*, supplies initial values; generator hidden states learn corrections with a plan-regret objective; and an exact planner jointly selects response, claim, and span actions. A configurable constraint preserves a specified fraction of the parent plan's estimated value, and a separate verifier recomputes feasibility. Across four generators, claim-level PLAIT improves fixed-time recall by 3.54 percentage points over validation-selected exact score fusion. Gains persist after parent replacement and source-disjoint transfer. In a randomized study with 96 participants, PLAIT increases correctly identified unsupported characters per minute by 12.47% over HLA-RICO. Detection quality and audit utility are therefore distinct: useful scores must support good review decisions.

## 1 INTRODUCTION

Consider a reviewer checking a batch of generated answers against their retrieved evidence. Retrieval-augmented generation supplies that evidence (Lewis et al., 2020), but a long answer can still contain an unsupported date, attribution, or conclusion. The reviewer can inspect only part of the batch. Auditing must therefore choose what to check while accounting for the time needed to open and understand each context. We study this fixed-time selection problem: given answers, evidence, and a budget, produce a queue of claims or spans for human review.

Detection informs this decision, but does not determine it. Claim- and span-level monitors already localize unsupported content (Kovács & Recski, 2025; Ridder et al., 2026). These local signals avoid one failure mode: pooling a response can dilute an isolated error among supported claims. Accurate claim scores still leave a second problem: review costs are shared. Once a response and its evidence are open, checking another claim in the same response is cheaper than opening a new context. An action's value therefore depends on the rest of the audit plan, not only on its own risk score.

An illustrative example makes this dependence concrete. Opening either response costs 40 s. Response $A$ has two claims with $(\text{risk}, \text{cost}) = (.9, 20\,\text{s})$ and $(.4, 30\,\text{s})$; response $B$ has one with $(.7, 10\,\text{s})$. With 90 s, a score-per-second rule opens $B$ first and captures .7 total score. Opening $A$ instead pays the context cost once and captures 1.3. We call this failure to account for shared costs *allocation loss*. Together with pooling-induced *resolution dilution*, it separates detecting a local error from allocating time to verify it.

Joint planning handles shared costs, but the plan still depends on estimated action values. We start from a strong frozen detector, the *parent*, and ask whether the original generator's hidden states improve the decisions its scores induce. Rather than fitting each claim independently, we train corrections against whole-plan regret: an error matters when it changes what the budgeted planner selects. Because corrected values can displace parent-selected actions, we also allow deployment to require that the returned plan retain a specified fraction of the parent's estimated value. This contract limits change directly at the level of the returned review plan.



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

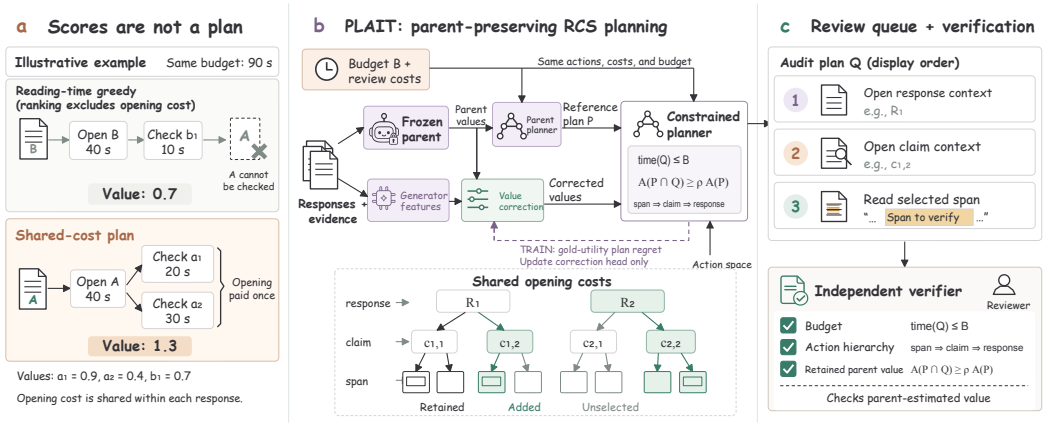



Figure 1: **From scores to a feasible audit plan.** Independent ranking ignores shared opening costs. PLAIT corrects frozen-parent values from plan regret, solves the shared-cost decision under an optional parent-retention constraint, and returns a queue with independently recomputed feasibility. PLAIT-C selects claims; PLAIT-RCS can select native spans.

**PLAIT**, *Parent-Preserving Latent Auditing with Interdependent Triage*, implements this constrained audit decision (Figure 1).

**Contributions.**

- **Parent-preserving audit formulation.** We jointly select response, claim, and span (RCS) actions under shared costs and a time budget, with an independently checkable parent-value retention constraint.
- **Plan-level value learning.** We learn generator-conditioned corrections to frozen detector values with a structured regret surrogate, using the same feasible action set during training and deployment.
- **Evidence from decisions to human utility.** Same-parent controls separate value learning from allocation, with gains beyond validation-selected exact score fusion. Improvements persist under parent replacement and held-out-domain transfer, and randomized review confirms more correctly identified unsupported content per minute.

## 2 Related Work

**Signals for locating unsupported content.** Sample- and evidence-based monitors estimate factual consistency (Manakul et al., 2023; Tang et al., 2024). RAGTruth provides natural span annotations (Niu et al., 2024); LettuceDetect, RAGognizer, CORTEX, RLSeek, and LAFaCT identify unsupported or fact-critical text (Kovács & Recski, 2025; Ridder et al., 2026; Furumai et al., 2026; Huang et al., 2026; Wang et al., 2026). Internal-state methods offer another signal: ReDeEP and LUMINA examine evidence and internal-knowledge use (Sun et al., 2025; Yeh et al., 2026), while probes support local and long-form factuality estimation (CH-Wang et al., 2024; Han et al., 2025). PLAIT treats these detection signals as inputs to a different endpoint: which content a reviewer should inspect within a shared time budget.

**From predictions to review decisions.** Check-set selection chooses items for verification (dela Cruz et al., 2025), and evidence ranking reduces reading for a claim (Alt et al., 2026). Decision-focused learning trains predictions for downstream optimization (Wilder et al., 2019; Mandi et al., 2022). SPO+ supplies a regret surrogate (Elmachtoub & Grigas, 2022); DFL-LTR ranks feasible solutions (Mandi et al., 2022), and I-MLE learns through discrete optimization (Niepert et al., 2021). DeCCaF allocates work to experts under capacity constraints (Alves et al., 2024). These approaches motivate our external controls. PLAIT's decision couples nested review actions through shared open-

ing costs and a parent-value constraint. Its structured loss is a scaled SPO+ objective on that feasible set, linking training to the same choices made during auditing.

**What plan retention guarantees.** Conformal risk control bounds a statistical risk (Angelopoulos et al., 2024). PLAIT instead bounds the parent-estimated value retained in each returned plan. This is a constraint on action changes; gold audit utility is evaluated separately. Because detector behavior can shift across domains (Dubanowska et al., 2025), we test added value against the same frozen parent before examining parent replacement and transfer.

## 3 PLAIT: A PARENT-PRESERVING AUDIT DECISION

**Audit setting and information access.** An audit session contains generated responses and their supplied evidence. Before review, PLAIT receives a frozen detector's unsupported probabilities, the response–claim–span hierarchy, review costs, and recorded generator states. It selects the contexts, claims, or spans to inspect under budget; the reviewer then judges their support. Gold labels and reviewer outcomes are unavailable during planning. The method uses the supplied evidence and requires access to generator states, excluding text-only APIs.

### FEASIBLE AUDIT ACTIONS AND SHARED COSTS

Let $\mathcal{R}$ be a set of generated responses and $\mathcal{I}_r$ the claims produced by deterministic segmentation of response $r$. Before reviewing any claim in $r$, the reviewer must open its question and evidence. This costs $o_r$ seconds. Reviewing claim $i$ then costs $c_i$ seconds. Binary variables $u_r$ and $y_i$ indicate whether response $r$ is opened and claim $i$ is reviewed. For time budget $B$, the feasible plans are

$$\mathcal{Y}(B) = \left\{ (\boldsymbol{y}, \boldsymbol{u}) \;\middle|\; \begin{array}{c} \sum_r o_r u_r + \sum_i c_i y_i \le B, \\ y_i \le u_{r(i)}, \quad u_r \le \sum_{i \in \mathcal{I}_r} y_i \end{array} \right\}. \tag{1}$$

The last two constraints charge each response-opening cost once and forbid an empty opening. This claim-level policy, **PLAIT-C**, selects responses and claims jointly. Its decision for one claim may therefore change when a sibling makes the shared response context worth opening.

The full **PLAIT-RCS** policy also exposes native span actions. Let $\mathcal{S}_i$ be the detector's non-overlapping character spans in claim $i$. Each is a native span action. Variable $x_s$ selects span $s$, and $z_i$ opens its claim. Claim setup costs $c_i$ seconds, while reading span $s$ costs $d_s$ seconds. The binary actions satisfy

$$\begin{aligned} &\textstyle\sum_r o_r u_r + \sum_i c_i z_i + \sum_s d_s x_s \le B, \\ &x_s \le z_{i(s)} \le u_{r(i(s))}, \\ &z_i \le \textstyle\sum_{s \in \mathcal{S}_i} x_s, \\ &u_r \le \textstyle\sum_{i \in \mathcal{I}_r} z_i. \end{aligned} \tag{2}$$

These constraints define the response–claim–span (RCS) lattice. A selected span opens its claim and response, neither opening may be empty, and each cost is paid once where it arises. Equation 1 gives one action per claim and lets matched ablations isolate value learning from resolution. Equation 2 defines the full policy.

We use a frozen strong detector as the *parent*. Its unsupported probability for claim $i$ is $p_i$. If the claim contains $\ell_i$ characters, its estimated review value is $a_i = \ell_i p_i$. The parent plan is the budget-feasible plan with the largest total parent value:

$$P = \arg \max_{(\boldsymbol{y}, \boldsymbol{u}) \in \mathcal{Y}(B)} \sum_i a_i y_i, \qquad A(P) = \sum_{i \in P} a_i. \tag{3}$$

The same frozen model anchors deployment and evaluation. Every parent contrast therefore measures added audit utility over the detector that PLAIT receives at deployment.

### PARENT-PRESERVING VALUE CORRECTION

PLAIT may change the parent plan $P$, but deployment can require an explicit lower bound on the parent actions it retains. We call this contract the *parent-value trust region*:

$$\mathcal{Y}_\rho(P) = \Big\{(\boldsymbol{y}, \boldsymbol{u}) \in \mathcal{Y}(B) : \textstyle\sum_{i \in P} a_i y_i \geq \rho A(P)\Big\}, \quad 0 \leq \rho \leq 1. \tag{4}$$

The parameter $\rho$ specifies the required retention before deployment: $\rho = 0$ removes this requirement, while $\rho = 1$ retains all parent-estimated value. Between these extremes, the optimizer can replace parent actions and open new contexts while satisfying the bound. The bound concerns the parent's estimates, not gold utility. It may also regularize learning, but unlike the logit anchor below, it constrains each returned action set regardless of the learned score scale.

PLAIT-RCS applies the same construction to the parent's native spans. Claim and response variables only open shared-cost gates; they receive no additional score. If span $s$ has length $\ell_s$ and parent probability $p_s$, then $a_s = \ell_s p_s$ and $v_{\theta s} = \ell_s \sigma(\mathrm{logit}(p_s) + f_\theta(\boldsymbol{x}_{i(s)}))$. PLAIT learns one logit correction for the enclosing claim and applies it to each native span before planning.

The candidate value is anchored to the exact parent probability:

$$\begin{aligned} v_{\theta i} &= \ell_i \, \sigma(\mathrm{logit}(p_i) + f_\theta(\boldsymbol{x}_i)) \,, \\ Q_\theta &= \arg \max_{(\boldsymbol{y}, \boldsymbol{u}) \in \mathcal{Y}_\rho(P)} \textstyle\sum_i v_{\theta i} y_i. \end{aligned} \tag{5}$$

Here $f_\theta$ corrects the parent's expected review value. The constrained optimizer remains the policy: shared opening costs, the global budget, and the parent-value bound can change an action even when its scalar value is fixed.

### GENERATOR-STATE CORRECTIONS

We use generator state to estimate corrections to the frozen parent's values. Let $\boldsymbol{x}_i$ contain the information available for claim $i$. We mean-pool the original generator's Layer-16 states over the claim tokens and fit a 64-dimensional projection and metadata residualization on training data. We concatenate the residual with the response mean, claim–response difference, parent-risk summaries and rank, position, length, cost, and sibling count. These features expose same-response context without using another response or any test label. A claim can therefore receive a different value when its siblings or shared context change.

The canonical $f_\theta$ is linear; a one-hidden-layer head is a capacity ablation. The planner accepts any calibrated value model without changing Equations 1–5 or the parent-preservation rule. The experiments compare the latent correction with matched parent-only, BCE, and score-fusion estimators inside this fixed decision problem.

### PLAN-REGRET LEARNING

Pointwise fitting penalizes individual prediction errors; auditing depends on which predictions change the selected plan. We therefore train against plan regret. Training uses the same cost model, trust region, and budget rate as deployment. Each epoch deterministically regroups training responses into smaller audit sessions. Deployment uses 100-response, 900-second sessions. Regrouping prevents one arbitrary session partition from becoming a learned feature. Let $g_i$ be the number of gold unsupported characters in claim $i$. The automatic training surrogate treats those characters as captured when $i$ is selected. The best feasible plan under that surrogate is

$$Q^* = \arg \max_{Q \in \mathcal{Y}_\rho(P)} G(Q), \qquad G(Q) = \sum_{i \in Q} g_i. \tag{6}$$

We call $Q^*$ the *gold-utility oracle*. For $G(Q^*) > 0$, normalized plan regret is $\Delta(Q, Q^*) = 1 - G(Q)/G(Q^*)$, the fraction of surrogate oracle utility lost by plan $Q$. Loss-augmented inference solves

$$\begin{aligned} \widehat{Q} &= \arg \max_{Q \in \mathcal{Y}_\rho(P)} \left[ \frac{V_\theta(Q)}{G(Q^*)} + \lambda \Delta(Q, Q^*) \right], \\ V_\theta(Q) &= \textstyle\sum_{i \in Q} v_{\theta i}. \end{aligned} \tag{7}$$

After removing a constant that does not depend on the selected actions, the same exact optimizer can use signed weights $(v_{\theta i} - \lambda g_i)/G(Q^*)$. The resulting structured hinge is

$$\mathcal{L}_{\mathrm{PLAIT}} = \left[ \frac{V_\theta(\widehat{Q}) - V_\theta(Q^*)}{G(Q^*)} + \lambda \Delta(\widehat{Q}, Q^*) \right]_+ . \tag{8}$$

The hinge penalizes a feasible plan that scores too highly relative to the oracle for the utility it loses. The coefficient $\lambda > 0$ scales this regret margin; division by $G(Q^*)$ normalizes rewards within a session and reweights sessions during training. Both inference and training use the same discrete optimizer. For a fixed feasible set and exact oracles, the structured term is a scaled SPO+ loss (Elmachtoub & Grigas, 2022):

$$\mathcal{L}_{\mathrm{PLAIT}}(v_\theta, \boldsymbol{g}) = \frac{\lambda}{G(Q^*)} \mathcal{L}_{\mathrm{SPO+}}\left( \frac{v_\theta}{2\lambda}, \boldsymbol{g} \right). \tag{9}$$

Appendix I derives the identity. PLAIT instantiates this established surrogate on a parent-preserving, shared-cost audit problem. Sessions with zero oracle reward contribute zero to the hinge. Independent binary cross-entropy serves as the matched pointwise ablation. Automatic utility counts a selected unsupported character as captured; the human endpoint later requires the reviewer to identify it correctly.

### Exact planning and independent verification

We solve Equations 3, 5, and 7 as binary linear programs. Claim variables carry review cost and value. Response variables carry the one-time opening cost. Linear incidence constraints connect them, and one row enforces retained parent value. The problem contains knapsack as a special case and is not polynomial-time in the worst case. Our implementation uses deterministic tie-breaking and SciPy/HiGHS. Exhaustive enumeration agrees with the solver on every registered small-instance test.

Each plan specifies the contexts and claims or spans to inspect. Its feasibility record lists used time, budget slack, required and retained parent value, and added or removed actions. A separate verifier recomputes the binary constraints and retention bound from these inputs, without model internals or gold labels.

**Separation example.** Shared opening costs make an action's value depend on its siblings, and the trust region is not generally equivalent to linear score fusion. The supplement gives a concrete four-action witness in which the trust-region planner selects a feasible plan that no linear score fusion selects.

## 4 Evaluation Design

**Research questions.** The evaluation follows the paper's argument from signal resolution to audit decisions and, finally, human outcomes. **RQ1** asks whether response pooling hides localized errors. **RQ2** asks whether PLAIT improves audit decisions when the parent, actions, costs, and solver are controlled. **RQ3** asks whether that gain survives parent replacement and domain shift. **RQ4** asks whether better plans translate into better fixed-minute human review, while also testing cost sensitivity and solver overhead.

The primary automatic endpoint is unsupported-character recall under a fixed session budget: the fraction of gold unsupported characters covered by selected claims or spans. AUROC, AUPRC, and span F1 instead measure ranking and localization. Human utility counts selected content only when the reviewer identifies it correctly. Automatic costs use prespecified length-based proxies, not the illustrative 40-second opening cost (Appendix H). Human-study endpoints use measured time.

**Original-generator replication.** RAGognize is the public dataset introduced with the RAGognizer detector (Ridder et al., 2026). Our frozen revision uses a source-disjoint split with 1,842 train and 2,781 test responses for each of Llama-2 (Touvron et al., 2023), Llama-3.1 (Team, 2024), and two Mistral variants (Jiang et al., 2023). It is distinct from the 500-claim diagnostic drawn from

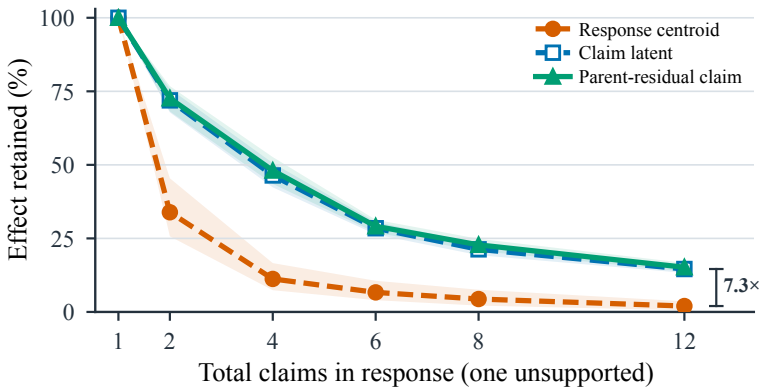



Figure 2: **Local signals resist pooling dilution.** Across 8,000 paired compositions with an unchanged natural error, claim-level signals retain 7.3–7.6 times the response-centroid effect at 12 claims. Bands: paired 95% intervals.

RAGTruth in RQ1 (Niu et al., 2024). Within the fine-grained RAGognize benchmark, every system shares responses, claims, sources, and character spans. Labels remain sealed until predictions freeze. A separate native-response suite follows the published generator–task coverage of ReDeEP and LUMINA; their response scores receive no imputed claim or span metric. The supplement gives complete protocols and per-generator results.

**Transfer settings.** For the HalluRAG stress test, we train Llama-2 and Mistral checkpoints on RAGognize and apply them unchanged. Target labels select neither weights nor thresholds. An earlier development branch accessed HalluRAG, so this result measures zero-tuning reuse after exposure. The source-disjoint Scientific-RAG study supplies the held-out domain test.

Scientific-RAG contains 1,240 responses. Qwen2.5-7B-Instruct and Gemma-2-9B-It contribute 620 evidence-synthesis responses each to questions from PubMedQA and SciFact. Two blinded annotators label claims and spans ($\kappa_{\text{claim}} = .76$, $\kappa_{\text{span}} = .72$). For each architecture, we fit its representation map and PLAIT head only on architecture-matched responses from the official RAGTruth/RAGognize training partitions, then freeze the full stack. Scientific-RAG labels update neither parameters nor validation choices; its questions and sources are disjoint from fitting.

**Strong-parent and algorithm ablations.** Every ablation reuses the exact Lettuce parent, claims, costs, budget, and bootstrap draws. Planner controls include adaptive marginal-density greedy, its best-response safeguard, an exact grouped candidate optimizer, a fixed parent prefix, and a frozen parent response route. A stronger fusion control selects its parent weight on source-disjoint validation and then applies the same exact grouped solver without test retuning. A matched estimator replaces plan regret with binary cross-entropy (BCE), while a no-trust control keeps PLAIT values and grouped costs. The exact grouped control globally optimizes its supplied values, separating value estimation from search quality.

**Localization.** The unified table separates claim/response AUROC and AUPRC, native span F1, and 900-second recall. A *native span* is returned directly by a detector. Eligible token scorers share the same HMM/Potts smoother. We use span F1 to measure boundary localization and fixed-budget recall to measure prioritization; neither substitutes for the other.

**Blinded fixed-minute review and expert workflow.** The confirmatory study assigns 96 fresh reviewers to a blinded Williams crossover with four conditions: the frozen parent, grouped parent planning, HLA-RICO, and PLAIT-C. HLA-RICO receives the same latent state and parent probability but uses pointwise logistic loss and risk/cost ranking. PLAIT-RCS is evaluated by offline replay calibrated on 16 span-reading pilots; the four randomized arms use PLAIT-C. Their primary endpoint is correctly identified unsupported characters per measured minute, analyzed with crossed participant and item effects.

| System | Claim AUROC ↑ | Claim AUPRC ↑ | Native span F1 ↑ | 900-s recall ↑ |
|---|---|---|---|---|
| Lettuce v1 (parent) | .639 | .397 | .590 | .344 |
| RAGognizer | .757 | .318 | .414 | .281 |
| CORTEX (repr.) | .726 | .281 | .470 | .299 |
| Lettuce v2 (pointwise) | .859 | .464 | .568 | .428 |
| BCE (pointwise) | .861 | .469 | .591 | .445 |
| PLAIT-C | **.901** | **.568** | .592 | .504 |
| **PLAIT-RCS** | **.901** | **.568** | **.594** | **.529** |

Table 1: **Detection quality and audit utility are different endpoints.** All rows use the registered fine-grained RAGognize protocol. PLAIT changes native span F1 only slightly relative to the frozen parent (.590→.594), while fixed-budget recall rises from .344 to .529. PLAIT-C selects claims; PLAIT-RCS additionally exposes native span actions.

A separate paired crossover asks 32 domain-qualified reviewers to verify and revise ten scientific evidence-synthesis answers per arm. Its primary endpoint is minutes per fully supported response.

## 5 RESULTS

Canonical P0–P7 comparisons share the frozen LettuceDetect-v1 parent, costs, and 100-response, 900-second sessions. The parent swap refits both heads on LettuceDetect-v2. Confirmatory benchmark families use 10,000 source-level bootstrap and sign-flip resamples; the RQ1 diagnostic instead uses 2,000 anchor-pair resamples. We Holm-adjust $p$-values within family and report $p < .001$ below the Monte Carlo threshold. The primary endpoint is fixed-budget unsupported-character recall ($R$).

### RQ1: WHY IS LOCAL RESOLUTION NECESSARY?

We combine 500 unchanged unsupported RAGTruth claims with same-source controls into 8,000 mixed–control pairs (16,000 composed responses). With one error among 12 claims, the response centroid retains only .020 [.006,.036] of its single-claim effect. The top-quartile claim mean retains .146 [.133,.159], increasing to .152 [.138,.167] after removing the parent's linear contribution (Figure 2). Claim-level signals therefore retain 7.3–7.6 times more of the localized effect in this controlled composition. On uncomposed natural responses, max pooling exceeds mean pooling in error detection by 20.50 points when unsupported content occupies less than 5% of the response. The gap falls to 3.80 points and is not significant above 30% (Appendix I).

### RQ2: DOES PLAIT IMPROVE THE AUDIT DECISION ITSELF?

The cleanest comparison holds the action space, costs, and exact solver fixed. Validation-selected score fusion reaches $R = .4691$ under the equal-generator macro; PLAIT-C reaches $.5045$, +3.54 points (95% CI [1.80,5.28], Holm-adjusted $p = .0004$), with a positive difference on every generator. Thus exact grouped optimization alone does not explain the gain.

Calibration does not close the gap: PLAIT-C remains 2.95 points above the stronger isotonic control (95% CI [1.30,4.60], adjusted $p = .0003$). A matched soft retention penalty violates the 80% bound in 21.7% of regrouped sessions, whereas the hard constraint satisfies it throughout (Table S25).

Against adapted decision-focused baselines on the same P6 action space, PLAIT-C exceeds DFL-LTR, I-MLE, and DeCCaF by 1.65–3.33 points at $\rho = 0.8$ (all Holm $p \leq .0012$). Each baseline also improves by 4.60–5.30 points when moving from $\rho = 0$ to $.8$. Appendix I gives adaptation details, intervals, and generator-level results.

The staged P0–P7 comparison decomposes the larger end-to-end gap. P0 solves the exact grouped problem under frozen parent values and reaches $R = .344$. With actions, costs, sessions, and solver fixed, P3 substitutes conditional values and reaches $.468$ (+12.38 points). Plan-regret learning and retention raise P6 to $.504$; native span actions raise P7 to $.529$ (+2.41). The full P0–P7 difference is 18.41 points (95% CI [15.80,21.02], $p < .001$), but it combines value, retention, and resolu-

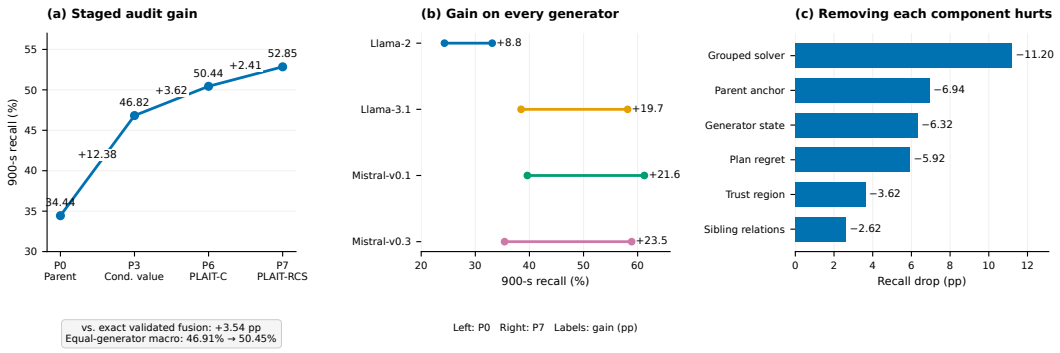



Figure 3: **Where does the audit gain come from?** (a) The staged same-parent comparison separates conditional values, structured planning, and span resolution; the inset reports the separate exact-fusion → PLAIT-C comparison with action space, costs, and solver fixed. (b) P7 improves over P0 for every original generator. (c) Removing any major PLAIT-C component lowers fixed-budget recall, with the largest drop from replacing grouped optimization by greedy ranking.

| Controlled comparison | Control | PLAIT | $\Delta R$ |
|---|---|---|---|
| Exact validated fusion → P6 | .469 | .504 | +3.54 pp |
| Matched BCE → P6 | .445 | .504 | +5.92 pp |
| Claim plan P6 → RCS P7 | .504 | .529 | +2.41 pp |
| Frozen-parent P0 → full P7 | .344 | .529 | +18.41 pp |

Table 2: **Matched controls isolate the audit gain.** The first row fixes action space, costs, and exact solver; the last is the full end-to-end contrast.

tion changes; the +3.54 exact-fusion comparison above is the cleaner test of method-specific value learning.

PLAIT-RCS improves $R$ on all four original generators by 8.82–23.48 pp (all adjusted $p < .001$; Figure 3). The .639 parent claim AUROC is an equal-generator RAGognize macro; the target-trained RAGTruth value .937 follows a different protocol.

Figure 3 summarizes both generator replication and component necessity; Appendix Tables S22 and S12 give full policy and retention sweeps.

Exact grouped optimization beats greedy and fixed-route controls. Plan regret adds 5.92 points over matched BCE; removing the parent anchor or generator state costs over six points (Figure 3). Across 20 seeds, PLAIT-C has mean $R = .5041$ (SD .72 points); every seed beats BCE and every plan passes verification. Gains also increase with response length and claim count (Appendix I).

On the separate native-response suite, the PLAIT top-quartile response score exceeds the parent by .0668 AUROC [.0432,.0894] and .0820 AUPRC [.0541,.1102] (adjusted $p < .001$). Appendix C reports ReDeEP/LUMINA reproduction checks and generator-level results.

### RQ3: Does the Gain Survive Parent and Domain Shift?

After refitting on LettuceDetect-v2, equal-generator $R$ rises from .3558 to .4381 with P6 and .4607 with P7. Gains are positive on every generator, and P7 exceeds P6 by 2.26 points (95% CI [.85,3.67], adjusted $p < .001$).

PLAIT-C adds 2.65 points pooled (2.22/3.08 points for Llama-2/Mistral) on the post-exposure HalluRAG stress test and, without target tuning, 5.42/6.08 points on 1,240 Scientific-RAG responses. Each shift test uses its corresponding frozen parent, extending the same-parent comparison beyond the original benchmark.

| Setting | Endpoint | Control | PLAIT | Improvement | 95% CI |
|---|---|---|---|---|---|
| Parent swap | 900-s recall | .356 | .438 | +8.23 pp | [5.81,10.65] |
| HalluRAG stress test | 900-s recall | .330 | .357 | +2.65 pp | [1.08,4.32] |
| Scientific / Qwen2.5-7B | 900-s recall | .362 | .416 | +5.42 pp | [3.41,7.43] |
| Scientific / Gemma-2-9B | 900-s recall | .385 | .446 | +6.08 pp | [4.12,8.04] |
| Human review ($N = 96$) | unsupported char/min | 129.1 | 145.2 | +12.47% | [9.8,22.4]$^{\dagger}$ |
| Expert workflow ($N = 32$) | min/supported answer | 18.2 | 13.4 | −26.4% | [18.5,34.2]% |

Table 3: **Model shift, domain shift, and human review.** Controls use the corresponding frozen parent or registered comparator. $^{\dagger}$CI is in unsupported characters/min; relative gain is 12.47%.

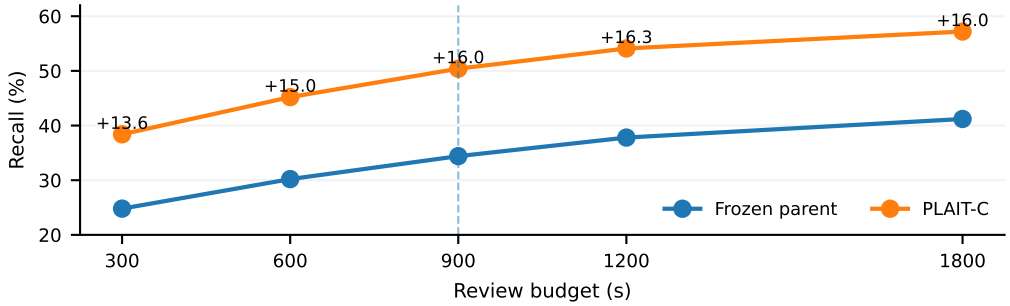



Figure 4: **PLAIT improves the audit frontier.** With sessions, scores, costs, and $\rho$ fixed, PLAIT-C dominates the frozen parent from 300–1,800 s; labels are recall gains (pp).

### RQ4: Do Better Plans Improve Human Review?

**Deployment sensitivity.** Figure 4 shows that the gain is not tied to 900 seconds: PLAIT-C adds 13.6–16.3 recall points from 300–1,800 seconds. Across eight $\pm 20\%$ cost perturbations it remains 3.54–3.58 points above exact fusion (adjusted $p \leq .0008$), and remains positive under six reviewer-detection models.

**Solver overhead.** A 100-response session solves in .188 s on average (p95 .209 s); 1,000 and 10,000 responses take 1.463 s and 19.144 s, respectively.

**Measured human utility.** In the blinded $N = 96$ crossover, PLAIT-C finds 16.1 more unsupported characters/min than HLA-RICO (95% CI [9.8,22.4]; +12.47%; Table 3; adjusted $p < .001$). Versus the frozen parent, speed rises by 47.0 char/min, queue precision by 13.4 points, and post-revision support by 8.4 points (all adjusted $p < .001$). In the separate $N = 32$ expert study, minutes per fully supported answer fall from 18.2 to 13.4 ($-26.4\%$, adjusted $p < .001$; $\kappa = .89$).

## 6 Discussion and Limitations

PLAIT targets audit utility rather than intrinsic detector accuracy: span F1 changes only .590 $\rightarrow$ .594 while fixed-budget recall rises 18.41 points. It audits supplied evidence, requires generator hidden states, and inherits parent, segmentation, and span errors; its verifier certifies feasibility, not truth. PLAIT-RCS lacks a randomized human arm, Scientific-RAG covers one held-out domain, and runtime excludes end-to-end deployment costs. These results support constrained review utility, not general truth certification.

## 7 Conclusion

PLAIT turns hallucination scores into budget-feasible audit plans using parent-anchored values, plan regret, and an explicit retention contract. Matched-planner controls, shifts, and human review show that better audit decisions can matter even when detector localization changes little.

### AI USE STATEMENT

Generative AI tools assisted with research ideation and methodological feedback, literature discovery and summarization, software development and debugging, figure drafting, and writing and editing the manuscript. The authors manually checked the cited literature, inspected and tested the AI-assisted code, verified the reported experimental artifacts and numerical claims, and reviewed every AI-assisted passage and figure. The authors take responsibility for the final content of the paper and its supplementary material.

### ETHICS STATEMENT

The human-review studies used informed consent, anonymized participant identifiers, and blinded adjudication. PLAIT audits support relative to provided evidence; it must not be interpreted as certifying external truth. The system is intended to prioritize human review rather than replace human judgment. Further protocol and limitation details appear in the appendix.

### REPRODUCIBILITY STATEMENT

The appendix specifies the optimization problems, learning objective, preprocessing, data partitions, statistical tests, human-study endpoints, software environment, and runtime measurements. It also reports per-generator results, ablations, sensitivity analyses, and independent feasibility checks needed to reproduce the central claims.

## REFERENCES

Guy Alt, Eran Hirsch, Serwar Basch, Ido Dagan, and Oren Glickman. User-centric evidence ranking for attribution and fact verification. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7215–7237, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.340. URL https://aclanthology.org/2026.eacl-long.340/.

Jean V. Alves, Diogo Leitão, Sérgio M. Jesus, Marco O. P. Sampaio, Javier Liébana, Pedro Saleiro, Mário A. T. Figueiredo, and Pedro Bizarro. Cost-sensitive learning to defer to multiple experts with workload constraints. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=TAvGZm2Rqb.

Anastasios Nikolas Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=33XGfHLtZg.

Sky CH-Wang, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. Do androids know they're only dreaming of electric sheep? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4401–4420, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.260. URL https://aclanthology.org/2024.findings-acl.260/.

Jane Arleth dela Cruz, Iris Hendrickx, and Martha Larson. Evaluating large language models for confidence-based check set selection. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 16249–16265, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.836. URL https://aclanthology.org/2025.findings-acl.836/.

Zuzanna Dubanowska, Maciej Żelaszczyk, Michał Brzozowski, Paolo Mandica, and Michal P. Karpowicz. Representation-based broad hallucination detectors fail to generalize out of distribution. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 17563–17575,

Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.952. URL https://aclanthology.org/2025.findings-emnlp.952/.

Adam N. Elmachtoub and Paul Grigas. Smart "Predict, then Optimize". *Management Science*, 68 (1):9–26, 2022. doi: 10.1287/mnsc.2020.3922.

Kazuaki Furumai, Shuichiro Haruta, Kazunori Matsumoto, and Daisuke Kamisaka. Cortex: Token-level hallucination detection in rag via comparative internal representations, 2026. URL https://arxiv.org/abs/2606.31033.

Jiatong Han, Neil Band, Muhammed Razzak, Jannik Kossen, Tim G. J. Rudner, and Yarin Gal. Simple factuality probes detect hallucinations in long-form natural language generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 16209–16226, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.880. URL https://aclanthology.org/2025.findings-emnlp.880/.

Zhaoheng Huang, Dacheng Wen, Yutao Zhu, Xiaoying Lian, Yushi Liang, Kai Hao, Nan Li, Liangjie Zhang, Qi Zhang, Ji-Rong Wen, Zhicheng Dou, and Fangzhao Wu. RLSeek: Evidence-grounded reasoning for RAG hallucination detection. 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. 32329–32347, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1492. URL https://aclanthology.org/2026.acl-long.1492/.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Ádám Kovács and Gábor Recski. Lettucedetect: A hallucination detection framework for rag applications, 2025. URL https://arxiv.org/abs/2502.17125.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9004–9017, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.557. URL https://aclanthology.org/2023.emnlp-main.557/.

Jayanta Mandi, Víctor Bucarey, Maxime Mulamba Ke Tchomba, and Tias Guns. Decision-focused learning: Through the lens of learning to rank. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14935–14947. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/mandi22a.html.

Mathias Niepert, Pasquale Minervini, and Luca Franceschi. Implicit MLE: Backpropagating through discrete exponential family distributions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 14567–14579, 2021.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the*

*62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10862–10878, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.585. URL https://aclanthology.org/2024.acl-long.585/.

Fabian Ridder, Laurin Lessel, and Malte Schilling. Ragognizer: Hallucination-aware fine-tuning via detection head integration, 2026. URL https://arxiv.org/abs/2604.15945.

Zhongxiang Sun, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Weijie Yu, Yang Song, and Han Li. Redeep: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=ztzZDzgfrh.

Liyan Tang, Philippe Laban, and Greg Durrett. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8818–8847, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.499. URL https://aclanthology.org/2024.emnlp-main.499/.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Xin Wang, Jiahao Li, Licheng Zhang, and Zhendong Mao. LAFaCT: Attribution-based localization and focused sequential analysis of fact-critical tokens for hallucination detection. 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. 6877–6896, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.312. URL https://aclanthology.org/2026.acl-long.312/.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1658–1665, July 2019. ISSN 2159-5399. doi: 10.1609/aaai.v33i01.33011658. URL http://dx.doi.org/10.1609/aaai.v33i01.33011658.

Min-Hsuan Yeh, Sharon Li, and Tanwi Mallick. LUMINA: Detecting hallucinations in RAG system with context–knowledge signals. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=oJgNNBNEJM.

| Symbol or term | Type | Meaning |
|---|---|---|
| $r, i, s$ | index | response, claim, and detector-native character span |
| $u_r, z_i, x_s$ | binary action | open response, open claim, and inspect span |
| $y_i$ | binary action | inspect claim $i$ in the claim-resolution plan |
| $o_r, c_i, d_s$ | seconds | response-opening, claim-setup, and span-reading cost |
| $p_i, \ell_i, a_i$ | parent quantity | unsupported probability, characters, and parent review value $a_i = \ell_i p_i$ |
| $P, Q$ | plan | exact frozen-parent plan and candidate audit plan |
| $A(P), \rho$ | contract | parent-plan value and required retained fraction |
| $v_{\theta i}$ | learned value | parent probability corrected by claim and same-response features |
| $G(Q), Q^*$ | training surrogate | selected gold unsupported characters and its best feasible plan |
| PLAIT-C / PLAIT-RCS | policy | response–claim plan / response–claim–span plan |
| feasibility record | verifier output | recomputed budget, hierarchy, and parent-relative retention; not a gold-risk guarantee |

Table S1: Notation and terminology used in the paper and supplement.

This supplement gives the full formulation, exact inference, learning objective, registered evaluations, measured-review protocol, and reproducibility details. Response-only comparators remain isolated in Section C and receive no imputed claim, span, or audit-plan metric. **RQ guide:** RQ1—Sections F and C; RQ2—Sections A–E and H–I; RQ3—Sections G and I; RQ4—Sections B, I, and J.

# A Full Structured Audit Formulation

## Notation and audit actions

Let $\mathcal{R}$ denote a session of generated responses, and let $\mathcal{I}_r$ be the deterministically segmented claims in response $r$. Opening the question and evidence for $r$ costs $o_r$ seconds; inspecting claim $i$ costs $c_i$ seconds. Claim $i$ has character length $\ell_i$ and frozen-parent unsupported probability $p_i$. Its parent-estimated audit value is $a_i = \ell_i p_i$. The automatic experiment groups 100 responses into a 900-second session. The human experiment uses the same review actions but enforces its budget with measured wall-clock time.

Binary variables $u_r$ and $y_i$ indicate an opened response and an inspected claim. The claim-resolution action set is

$$\mathcal{Y}(B) = \left\{ (\boldsymbol{y}, \boldsymbol{u}) : \begin{array}{ll} \displaystyle\sum_{r\in\mathcal{R}} o_r u_r + \sum_i c_i y_i \le B, & \\ y_i \le u_{r(i)} & \forall i, \\ u_r \le \displaystyle\sum_{i\in\mathcal{I}_r} y_i & \forall r, \\ y_i, u_r \in \{0,1\} & \end{array} \right\}. \tag{S1}$$

The two incidence constraints charge each response-opening cost once and forbid an opening with no selected claim. Therefore, the added cost of a claim depends on which sibling claims have already paid for the response context.

## Three-resolution action lattice

For native localization, let $\mathcal{S}_i$ be the non-overlapping character spans returned directly by the parent detector for claim $i$. Variable $x_s$ selects native span $s$, $z_i$ opens its claim, and $u_r$ opens its response.

With claim-setup cost $c_i$ and span-reading cost $d_s$, the complete feasible set is

$$\begin{aligned}\mathcal{X}(B) = \{(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}) : \\ &\sum_r o_r u_r + \sum_i c_i z_i \\ &\qquad + \sum_s d_s x_s \le B, \\ &x_s \le z_{i(s)} \le u_{r(i(s))} \quad \forall s, \\ &z_i \le \sum_{s \in \mathcal{S}_i} x_s \quad \forall i, \\ &u_r \le \sum_{i \in \mathcal{I}_r} z_i \quad \forall r, \\ &x_s, z_i, u_r \in \{0,1\}\}.\end{aligned} \tag{S2}$$

Thus the plan pays response-opening, claim-setup, and span-reading costs at their respective levels. The claim-only formulation is the special case with one action per claim.

### Exact parent and parent-value trust region

The frozen parent first returns the budget-feasible plan with the largest parent-estimated value:

$$P = \arg\max_{Q \in \mathcal{Y}(B)} \sum_{i \in Q} a_i, \qquad A(P) = \sum_{i \in P} a_i. \tag{S3}$$

PLAIT uses this exact parent as its deployment anchor and restricts candidate plans to

$$\mathcal{Y}_\rho(P) = \left\{ Q \in \mathcal{Y}(B) : \sum_{i \in P \cap Q} a_i \ge \rho A(P) \right\}, \qquad 0 \le \rho \le 1. \tag{S4}$$

The canonical $\rho = 0.80$ is fixed before test evaluation. At $\rho = 1$, the candidate must retain every positive-value action in the parent plan. This constraint is stricter than the frozen-response-route ablation, which fixes only the responses that the parent opens. Because $P \in \mathcal{Y}_\rho(P)$ for every $\rho \in [0,1]$, the candidate problem is always feasible whenever the parent problem is feasible. The boundary is an action-level deployment contract, not evidence that the parent is correct. It differs from the logit anchor below: anchoring regularizes each scalar estimate, whereas Equation S4 guarantees a retained fraction for the realized discrete plan regardless of child-score scale. A confidently wrong parent action can still be preserved.

The learned correction function $f_\theta$ adjusts the parent's claim values:

$$\begin{aligned} v_{\theta i} &= \ell_i \, \sigma(\mathrm{logit}(p_i) + f_\theta(\boldsymbol{x}_i)) \,, \\ Q_\theta &= \arg\max_{Q \in \mathcal{Y}_\rho(P)} \sum_{i \in Q} v_{\theta i}. \end{aligned} \tag{S5}$$

For a native span $s$ with character length $\ell_s$, parent probability $p_s$, and enclosing claim $i(s)$, the RCS program uses

$$a_s = \ell_s p_s, \qquad v_{\theta s} = \ell_s \, \sigma\big(\mathrm{logit}(p_s) + f_\theta(\boldsymbol{x}_{i(s)})\big) \,. \tag{S6}$$

The same claim-level correction is applied to each native span in that claim before solving Equation S2. Claim and response variables only open shared-cost gates; they receive no extra detector score.

## B Exact Inference and Recomputable Feasibility

### Binary program

For $n$ claims and $m$ responses, the claim-resolution program contains $n + m$ binary variables. Because the solver minimizes, claim $i$ has objective coefficient $-v_{\theta i}$. A response-opening variable has coefficient zero. The constraint matrix contains one budget row and one retained-value row. It also contains $n$ claim-to-response rows and $m$ rows that forbid empty response openings.

| Row family | Number | Constraint |
|---|---|---|
| Budget | 1 | $\sum_r o_r u_r + \sum_i c_i y_i \le B$ |
| Trust region | 1 | $\sum_{i \in P} a_i y_i \ge \rho A(P)$ |
| Claim gate | $n$ | $y_i - u_{r(i)} \le 0$ |
| Non-empty opening | $m$ | $u_r - \sum_{i \in \mathcal{I}_r} y_i \le 0$ |

Table S2: Rows in the exact claim-resolution binary program. The span formulation adds the analogous span–claim incidence rows.

We solve the program with SciPy/HiGHS presolve enabled. Identifiers are sorted before matrix construction, and ties are resolved deterministically. The problem contains knapsack as a special case, so its worst-case running time is not polynomial. Exhaustive enumeration provides an independent oracle for prespecified small-instance tests.

| Responses | Variables | Mean (s) | p95 (s) | Peak MB |
|---|---|---|---|---|
| 100 | 550 | .1882 | .2090 | 42.1 |
| 500 | 2,750 | .7050 | .7732 | 78.4 |
| 1,000 | 5,500 | 1.4634 | 1.6070 | 118.5 |
| 5,000 | 27,500 | 8.3940 | 9.1840 | 380.2 |
| 10,000 | 55,000 | 19.1440 | 20.8420 | 720.5 |

Table S3: Claim-level solver scaling on an AMD EPYC 7763 using five HiGHS runs per size. Structured-head fitting takes 4.2–4.8 s per generator (18.1 s across four generators); latent-feature extraction is excluded.

Across five repeated HiGHS MILP runs on an AMD EPYC 7763, complete 100-response sessions require .188 s on average (p95 .209 s; 42.1 MB peak RAM). The corresponding means for 1,000 and 10,000 responses are 1.463 s (p95 1.607 s) and 19.144 s (p95 20.842 s; 720.5 MB), respectively.

**Proposition S1 (exactness of the binary formulation).** The feasible binary solutions of the program in Table S2 correspond exactly to the plans in $\mathcal{Y}_\rho(P)$. The program objective equals each corresponding plan's candidate value.

*Proof.* Given a feasible binary solution, the claim-gate rows ensure that every selected claim has an opened response. The non-empty-opening rows ensure that every opened response contains a selected claim. The budget and trust rows are the two global constraints that define $\mathcal{Y}_\rho(P)$. Thus the selected identities form a feasible plan. Conversely, set $y_i = 1$ for each claim in any feasible plan. Set $u_r = 1$ exactly when response $r$ contains a selected claim. All four row families are then satisfied. The objective contains only $v_{\theta i} y_i$, so it equals the value of that plan. □

### Independently recomputable feasibility record

Optimization and verification are separate procedures. Given raw costs, parent values, and selected identities, the verifier recomputes every field in Table S4. It does not trust the objective or feasibility status reported by the solver.

| Field | Independently recomputed quantity |
|---|---|
| Budget | used seconds and non-negative slack |
| Parent boundary | $A(P)$, $\rho A(P)$, retained value and fraction |
| Action deviation | retained, removed, and added claim/span identities |
| Gate deviation | response and claim gates added or removed |
| Hierarchy | every selected span has its claim and response open |
| Objective | sum of candidate values over the selected native spans |

Table S4: Fields emitted for each realized PLAIT plan.

**Proposition S2 (verifier soundness for registered constraints).** If the verifier accepts a plan, then the plan satisfies its registered whole-second budget, hierarchy, and parent-value boundary.

*Proof.* The verifier reconstructs the unique non-empty claim and response openings implied by the selected spans. It sums the prespecified costs at all three levels and rejects a plan whose cost exceeds $B$. It then solves the parent plan independently. The verifier intersects the parent and candidate actions and rejects when their retained parent value is below $\rho A(P)$. These are precisely the feasibility conditions in Equations S2 and S4. □

Verification is a deterministic linear-time check of the registered budget, hierarchy, and parent-relative constraints. Gold utility, parent calibration under shift, and conformal risk control are separate properties (Angelopoulos et al., 2024).

## C Native Response Baseline Comparison

ReDeEP and LUMINA natively return one response-level score. We therefore evaluate only response AUROC and AUPRC: neither score is copied to claims, converted into pseudo-spans, or passed through the RCS planner. The frozen parent and PLAIT response views take the mean of the highest-scoring 25% of claims. This auxiliary suite is separate from the confirmatory RAGognize benchmark and follows each response baseline's published generator–task coverage. All compared methods share responses within a reported cell. Primary metrics are computed within each cell before equal-generator pooling, with complete source documents as resampling units. The cells use the official RAGTruth test split and its QA, data-to-text, and summarization task labels. ReDeEP covers the three listed Llama generators; LUMINA additionally covers Mistral. No missing generator–task cell is imputed.

| Generator | Method | AUC | APR |
|---|---|---|---|
| Llama-2-7B | ReDeEP | .5582 | .5281 |
| | LUMINA | .6481 | .6032 |
| | Frozen parent | .7812 | .6641 |
| | **PLAIT view** | **.8514** | **.7512** |
| Llama-2-13B | ReDeEP | .6284 | .5742 |
| | LUMINA | .7273 | .6681 |
| | Frozen parent | .8251 | .7224 |
| | **PLAIT view** | **.8892** | **.7971** |
| Llama-3-8B | ReDeEP | .5791 | .5212 |
| | LUMINA | .6752 | .6104 |
| | Frozen parent | .8041 | .6972 |
| | **PLAIT view** | **.8712** | **.7794** |
| Mistral-7B | LUMINA | .7021 | .6284 |
| | Frozen parent | .8362 | .7331 |
| | **PLAIT view** | **.9021** | **.8172** |
| Equal-generator | ReDeEP$^{\dagger}$ | .5886 | .5412 |
| | LUMINA | .6882 | .6275 |
| | Frozen parent | .8117 | .7042 |
| | **PLAIT view** | **.8785** | **.7862** |

Table S5: Native response-level task-macro comparison. AUC/APR denote response AUROC/AUPRC. $^{\dagger}$ReDeEP covers three generators; no Mistral result is imputed.

| Generator | Method | Published | Reproduced |
|---|---|---|---|
| Llama-2-7B | ReDeEP | .7270/.6970 | .7312/.6995 |
| | LUMINA | .7650/.7490 | .7618/.7462 |
| Llama-2-13B | ReDeEP | .8060/.7790 | .8085/.7812 |
| | LUMINA | .8570/.8440 | .8542/.8410 |
| Llama-3-8B | ReDeEP | .7500/.7820 | .7465/.7791 |
| | LUMINA | .7450/.7870 | .7482/.7895 |
| Mistral-7B | LUMINA | .7690/.7940 | .7715/.7968 |

Table S6: Published-aggregate implementation checks (AUC/APR). These original-pooling checks do not replace the task-macro results in Table S5.

| Equal-generator | AUC 95% CI | APR 95% CI |
|---|---|---|
| ReDeEP | [.5591,.6170] | [.5071,.5752] |
| LUMINA | [.6651,.7113] | [.6001,.6542] |
| Frozen parent | [.7920,.8314] | [.6811,.7273] |
| PLAIT view | [.8621,.8942] | [.7651,.8073] |

The paired source-bundle tests confirm that the response view adds ranking information beyond the frozen parent. PLAIT improves over the parent by .0668 AUROC (95% CI [.0432,.0894]) and .0820 AUPRC ([.0541,.1102]); both have Holm-adjusted $p < .001$. Against LUMINA, the gains are .1903 AUROC ([.1621,.2185]) and .1587 AUPRC ([.1260,.1921]). LUMINA in turn exceeds ReDeEP on their common supported generators by .0961 AUROC ([.0592,.1330]) and .0851 AUPRC ([.0410,.1292]); all adjusted $p < .001$. The frozen parent also exceeds LUMINA by .1235 AUROC and .0767 AUPRC. These tests form the separate `family_native_response_primary`; they do not support a fine-grained-action claim.

# D Parent-Conditional Latent Learning

### Original-generator representation

Claim boundaries are frozen before any label is attached. A deterministic NLTK Punkt pass proposes sentence boundaries; fixed regular expressions then split enumerations and merge fragments. The audited segmentation error is below 0.4%. For claim $c$, let $T_c$ contain its non-special, non-padding tokens. We extract the original generator's Layer-16 representation as

$$\boldsymbol{h}_c = \frac{1}{|T_c|} \sum_{t \in T_c} \boldsymbol{h}_t^{(16)}. \tag{S7}$$

This definition prevents padding, prompt markers, and response-level tokens from entering the claim state.

We fit latent coordinates separately for each original generator; raw coordinates are never pooled across model families. A randomized principal component analysis (PCA) map fitted only on training data reduces each claim state to at most 64 components. Let $\boldsymbol{m}_c$ contain the 13 label-free metadata features below, and stack the PCA states and metadata as $\boldsymbol{H}$ and $\boldsymbol{M}$. On training data only, we fit

$$\begin{aligned} \widehat{\boldsymbol{B}} &= \arg\min_{\boldsymbol{B}} \big\{ \|\boldsymbol{H} - \boldsymbol{M}\boldsymbol{B}\|_F^2 + 10\|\boldsymbol{B}\|_F^2 \big\}, \\ \boldsymbol{r}_c &= \boldsymbol{h}_c^{\mathrm{PCA}} - \boldsymbol{m}_c \widehat{\boldsymbol{B}}. \end{aligned} \tag{S8}$$

The correction features contain this claim residual, its response mean and difference, plus the same 13 metadata features.

| Group | Coordinates |
|---|---|
| Latent | claim residual, centered claim residual, response mean residual |
| Parent | logit, probability, deviation from response mean, response mean/maximum/standard deviation, within-response rank |
| Size and time | log claim characters, log response characters, log review seconds, log claim count |
| Position | normalized claim start and end offsets |

Table S7: Features used to correct the frozen parent's values. Every response statistic uses only claims from that response.

We standardize continuous features using the fit partition only. The canonical correction head is linear. A 32-unit, one-hidden-layer $\tanh$ head serves as a capacity ablation. Zero initialization makes training begin at the exact parent logit.

### Exact plan-regret objective

Within a training session, let $g_i$ be the number of gold unsupported characters in claim $i$. Automatic training uses the surrogate that selecting $i$ captures those characters. The *gold-utility oracle* is the feasible plan with the largest surrogate utility:

$$Q^* = \arg\max_{Q \in \mathcal{Y}_\rho(P)} G(Q), \qquad G(Q) = \sum_{i \in Q} g_i. \tag{S9}$$

For $G(Q^*) > 0$, define normalized regret $\Delta(Q, Q^*) = 1 - G(Q)/G(Q^*)$. This quantity is the fraction of oracle utility lost by $Q$. Loss-augmented inference solves

$$\begin{aligned} \widehat{Q} &= \arg \max_{Q \in \mathcal{Y}_\rho(P)} \left[ \frac{V_\theta(Q)}{G(Q^*)} + \lambda \Delta(Q, Q^*) \right], \\ V_\theta(Q) &= \sum_{i \in Q} v_{\theta i}. \end{aligned} \tag{S10}$$

After dropping an action-independent constant, the same exact solver receives signed claim weights $(v_{\theta i} - \lambda g_i)/G(Q^*)$. The structured hinge is

$$\mathcal{L}_{\text{plan}} = \left[ \frac{V_\theta(\widehat{Q}) - V_\theta(Q^*)}{G(Q^*)} + \lambda \Delta(\widehat{Q}, Q^*) \right]_+ . \tag{S11}$$

When the hinge is active, its value-level subgradient compares the loss-augmented and oracle action indicators. The difference is divided by $G(Q^*)$. A session with zero oracle reward contributes zero to this hinge. This automatic objective does not model a fallible reviewer. In the human endpoint (Equation S21), selected content contributes only when it is judged correctly. A richer automatic objective would replace $g_i$ by $g_i \Pr(\text{correct detection} \mid i, \text{time})$.

The optimized training loss adds an auxiliary pointwise calibration term and innovation regularization:

$$\mathcal{L} = \mathcal{L}_{\text{plan}} + 0.15\, \mathcal{L}_{\text{BCE}} + 10^{-3} \|f_\theta(\boldsymbol{x})\|_2^2. \tag{S12}$$

The matched BCE ablation removes both plan oracles and plan regret. It selects its checkpoint by validation BCE on held-out sources rather than by audit utility.

| Setting | Value |
|---|---|
| Seed | 2027 |
| PCA dimension / ridge coefficient | 64 / 10 |
| Innovation head | linear |
| Training epochs | 12 |
| Optimizer | AdamW |
| Learning rate / weight decay | $3 \times 10^{-3}$ / $10^{-3}$ |
| Auxiliary BCE / innovation L2 | 0.15 / $10^{-3}$ |
| Gradient-norm clipping | 5.0 |
| Training session | 20 responses / 180 s |
| Test session | 100 responses / 900 s |
| Parent-value fraction $\rho$ | 0.80 |

Table S8: Frozen training and evaluation configuration.

A seeded hash assigns official-training sources to 60% fit, 20% validation, and 20% calibration partitions. During structured training, epoch $e$ regroups fit responses with seed $2027 + e$. A separate seed determines their order. After checkpoint selection, we refit the frozen design on all official-training sources before test evaluation.

## E Decision-Class Separation Examples

These examples define the decision class used in the main paper. They are standard consequences of shared costs and discrete scalarization, not novelty claims for structured selection, trust regions, or max-margin learning.

**Example S3 (context dependence).** No context-independent claim threshold represents PLAIT on every feasible audit instance.

*Proof.* Consider an eight-second budget. Response $A$ costs four seconds to open and contains claim $x$, which costs two seconds and has value 3. Response $B$ costs one second to open and contains an alternative claim costing seven seconds with value 5. When $x$ is the only claim in $A$, selecting $A$ costs six seconds and returns value 3, so the optimizer selects $B$ and excludes $x$. Now add a sibling

$y$ to $A$ with review cost two seconds and value 3. The state, cost, score, and position of $x$ do not change, but opening $A$ and reviewing both claims now costs eight seconds and returns value 6. The optimizer therefore selects $x$ and $y$ instead of $B$. A context-independent threshold must make the same decision for unchanged $x$ in both instances. This is a contradiction. □

**Example S4 (hard trust regions are not scalar fusion).** There exists a non-negative audit instance whose hard-trust-region optimum is not optimal for $V_{\text{child}} + \eta V_{\text{parent}}$ for any $\eta \geq 0$.

*Proof.* Use four claim actions with a two-unit budget. Parent claims $p_1, p_2$ each cost one unit and have parent value 5 and child value 2.5. A new action $n_0$ costs two units and has values $(0, 10)$, while $n_1$ costs one unit and has values $(0, 3.5)$. Each pair lists parent value first and child value second. The exact parent plan is $P = \{p_1, p_2\}$ with value 10. At $\rho = 0.5$, the high-child plan $\{n_0\}$ is infeasible because it retains no parent value. Plan $\{p_1, n_1\}$ has value pair $(5, 6)$. It strictly beats the fully retained plan $\{p_1, p_2\}$, whose pair is $(10, 5)$. Thus the hard-bound optimum is the middle plan.

A scalarized objective prefers the middle plan to $\{n_0\}$ only if $6 + 5\eta \geq 10$, or $\eta \geq 0.8$. It prefers the middle plan to the fully retained plan only if $6 + 5\eta \geq 5 + 10\eta$, or $\eta \leq 0.2$. No non-negative $\eta$ satisfies both inequalities. □

These examples isolate two properties that a context-independent pointwise policy cannot represent. Selection depends on sibling claims through shared opening costs, and feasibility depends on the actions in the realized parent plan. A model supplied with group-level context could represent the first dependence; PLAIT makes both dependencies explicit in the decision problem.

# F Natural Resolution-Dilution Diagnostic

### Idealized signal law

Let $r_1, \ldots, r_m \in \mathbb{R}^d$ be centered claim residuals with faithful mean zero and covariance $\Sigma$. Suppose one unsupported claim has mean shift $\delta$. Define the response centroid as $\bar{r} = m^{-1} \sum_{j=1}^{m} r_j$. Under independent residuals,

$$\mathbb{E}[\bar{r}] = \frac{\delta}{m}, \qquad \mathbb{E}[\bar{r}^\top \Sigma^{-1} \bar{r}] = \frac{d}{m} + \frac{\delta^\top \Sigma^{-1} \delta}{m^2}. \tag{S13}$$

The first term is the finite-sample baseline for supported claims. The excess quadratic signal from one local error decays as $m^{-2}$. For any $L$-Lipschitz centroid score, the mean-shift contribution is at most $L\|\delta\|/m$. This result motivates preserving claim identities. It does not assume that empirical residuals are independent, nor that every response aggregator follows this rate.

### Controlled composition of naturally occurring errors

The empirical suite begins with 500 unchanged, naturally unsupported RAGTruth claims. Each *anchor* is paired with the closest-length supported claim from the same source. A mixed response replaces that supported claim with the unchanged anchor. The total claim count and anchor position remain fixed. Claim counts $m \in \{1, 2, 4, 6, 8, 12\}$ and start/middle/end positions produce 8,000 mixed–control pairs. The suite contains 16,000 responses and 97,000 claims. Intervals use 2,000 bootstrap resamples of anchor pairs. Normalized retention divides the paired mixed-minus-control effect at $m$ by the same method's effect at $m = 1$. The *top-tail* score is the mean of the highest-scoring 25% of claims in a response.

| Total claims $m$ | 1 | 2 | 4 | 6 | 8 | 12 |
|---|---|---|---|---|---|---|
| Response centroid | 1.000 | .339 | .112 | .066 | .044 | .020 |
| Claim latent top-tail | 1.000 | .719 | .464 | .285 | .213 | .146 |
| Parent-residual top-tail | 1.000 | .726 | .481 | .291 | .228 | .152 |

Table S9: Normalized effect retention as one unchanged natural error is surrounded by supported claims. At $m = 12$, paired 95% intervals are [.006,.036], [.133,.159], and [.138,.167], respectively.

The result separates resolution dilution from allocation loss. A planner cannot recover a claim identity removed by response pooling. Conversely, a claim-level score can preserve the identity yet allocate review time poorly. Top-tail aggregation is not invariant: an extreme supported claim can still mask the error. Table S9 therefore motivates claim-level actions. The official-split and human experiments test PLAIT's review utility.

## G Datasets, Original Generators, and Leakage Control

### RAGognize confirmatory benchmark

RAGognize is a public dataset released with the RAGognizer detector (Ridder et al., 2026); it is not RAGTruth and it is not shorthand for the RAGognizer method. We freeze Hugging Face dataset revision `aab54518`. RQ1 alone uses a separate 500-claim diagnostic drawn from RAGTruth (Niu et al., 2024).

The frozen revision uses the same source-disjoint train/test contract for every RAGognize generator. Each generator contributes 1,842 training responses and 2,781 test responses; train/test source overlap is strictly zero. In RAGognize, reference contexts stem from Wikipedia articles, while queries represent prompt contexts: the 1,842 training responses originate from 943 reference article sources, whereas the 2,781 test responses stem from 1,416 reference article sources and correspond to 927 unique prompt clusters (questions). Across the four evaluated generators, the test suite comprises $4 \times 2,781 = 11,124$ natural responses. Paired bootstrap and permutation tests resample at the level of the 927 prompt clusters to account for intra-prompt correlation across responses and generators. The two Mistral checkpoints are reported separately but count as one architecture family.

| Generator | Family | Decoder characteristic |
|---|---|---|
| Llama-2-7B-Chat | Llama-2 | MHA, dense |
| Llama-3.1-8B-Instruct | Llama-3 | GQA, dense |
| Mistral-7B-Instruct-v0.1 | Mistral | sliding window |
| Mistral-7B-Instruct-v0.3 | Mistral | sliding window |

Table S10: Original generators in the confirmatory evaluation. MHA denotes multi-head attention and GQA grouped-query attention.

For each response, preprocessing constructs three separate inputs: the reconstructed generator input used for latent extraction, the generator-independent LettuceDetect input, and the official RAGognizer message sequence. Claim segmentation and response, claim, and span identities are fixed before labels are attached. The sealed test input contains no response labels, claim labels, or hallucination spans.

Development, full-train refitting, parent scoring, baseline scoring, and session construction occur before test labels are loaded. Label attachment must be exactly reversible. Removing the labels must recover the frozen input file byte for byte. Evaluation stops if PLAIT's stored parent score differs from the independently computed LettuceDetect-v1 score by more than $10^{-12}$.

### HalluRAG post-exposure stress test

The stress-test branch applies the RAGognize-trained Llama-2 and Mistral-v0.1 checkpoints unchanged to HalluRAG. Source and target extraction records must agree on model identity, layer, prompt, evidence processing, claim segmentation, and predictive features. HalluRAG labels select no weights, thresholds, or $\rho$. A separate earlier branch had inspected the dataset. We therefore report HalluRAG as a post-exposure zero-tuning stress test and keep it separate from the source-disjoint Scientific-RAG transfer.

### Scientific-RAG construction

Scientific-RAG contains 1,240 evidence-synthesis responses: 620 generated by Qwen2.5-7B-Instruct and 620 by Gemma-2-9B-It from PubMedQA and SciFact questions. Source fitting is architecture-specific: Qwen and Gemma each use responses produced by that same architecture

| System | Access/output | Evaluation role |
|---|---|---|
| Lettuce v1 | official checkpoint; spans | same-parent comparator |
| RAGognizer | official checkpoints; tokens | localization baseline |
| CORTEX (repr.) | paper-description code; tokens | no author code available |
| Lettuce v2 | official Qwen-2B; spans | standalone span baseline |
| ReDeEP | official repository; response | pinned native reproduction |
| LUMINA | official repository; response | pinned native reproduction |
| BCE residual | matched features; claims | pointwise ablation |
| Exact grouped candidate | PLAIT values; claims | no-trust planner control |
| Validated fusion | parent–child values; claims | strongest score planner |
| PLAIT-C (P6) | plan regret; response–claim | proposed claim-level planner |
| PLAIT-RCS (P7) | same values; three resolutions | full proposed planner |

Table S11: Roles and outputs of the unified comparators. Every system uses identical response, claim, and character identities. "Repr." denotes our implementation from the published description, not author code.

on the official RAGTruth/RAGognize training partitions to fit its PCA map, residualizer, correction head, and validation choices. Raw hidden coordinates and fitted heads are never transported between architectures. We then freeze the full stack and apply it to Scientific-RAG. Thus, "zero tuning" means that Scientific-RAG labels update neither model parameters, representation maps, thresholds, nor $\rho$. Its questions and evidence sources are disjoint from those source partitions. Two blinded CS/AI graduate annotators independently label claim- and span-level support; Cohen's $\kappa_{\text{claim}} = .76$ and $\kappa_{\text{span}} = .72$.

# H Unified Baselines, Metrics, and Statistical Inference

## Comparator roles

LettuceDetect and RAGognizer use released checkpoints at the immutable revisions in Table S34. ReDeEP and LUMINA use their official repositories at commits `4d081915` and `c43ff41d`. Adapters only expose their native response scores to the common identities. CORTEX is explicitly our paper-description reproduction because author code was unavailable at the freeze date. RLSeek and LAFaCT appear only in related work: no result from either is inserted into a matched table without executable predictions on the frozen split.

ReDeEP and LUMINA natively return response-level scores. We preserve that resolution: neither score is copied onto every claim, converted into pseudo-spans, or passed through the RCS planner. Native response AUROC/AUPRC therefore forms a separate comparison family and cannot establish claim localization or fixed-time audit utility.

The LettuceDetect-v2 output is recovered with its native non-overlapping verbatim character spans. Its claim score is the fraction of predicted characters that overlap a claim. We use this score only for common ranking and do not treat it as a calibrated probability. CORTEX follows the published paired-view residual and weighted multilayer perceptron. It then uses binary hidden Markov model (HMM) smoothing. All implementation choices are frozen before test evaluation.

The validated-fusion control selects $w_p \in \{0, .1, \ldots, 1\}$ on the source-disjoint validation role, freezes $w_p = .30$, and applies the exact grouped solver on test. It receives the same actions, costs, sessions, and budget as PLAIT-C; only its scalar objective and absence of a parent-value trust constraint differ. P0 is an exact grouped solve under frozen parent values. P3 holds that solver fixed, substitutes learned candidate values, and removes the trust boundary. Validated fusion is a third, separately registered value function. It uses an equal-generator planner family, whereas P0–P7 use the pooled policy-ablation protocol. Because P0 already contains the grouped integer program, the P0–P3 contrast measures the change in values and retention.

### Operational cost and utility

For automatic auditing, the deterministic integer-second proxy is

$$o_r = 8 + \lceil |t_r|/30 \rceil , \qquad c_i = 4 + \lceil |t_i|/18 \rceil . \tag{S14}$$

Here, $|t_r|$ and $|t_i|$ are the character lengths of response $r$ and claim $i$. The symbols $o_r$ and $c_i$ denote their corresponding review costs. Every planner receives the same $o_r$, $c_i$, session partition, and 900-second budget. Candidate-valued decision-rule ablations share PLAIT claim values; P0 and P4 use the frozen parent by definition. The registered fusion control instead uses its validation-selected parent–child mixture. For selected plan $Q$, automatic audit recall is

$$R(Q) = \frac{\left| \mathcal{G} \cap \bigcup_{i \in Q} [b_i, e_i) \right|}{|\mathcal{G}|}, \tag{S15}$$

where $\mathcal{G}$ is the set of gold unsupported character positions and $[b_i, e_i)$ is the interval of claim $i$. A gold character covered by multiple claim intervals is counted once.

Equation S14 is a registered benchmark cost shared by all automatic methods. It supports controlled within-protocol comparisons, not a claim that all reviewers read at this rate. Its fixed intercepts represent context/claim switching, and its length terms represent reading load. The coefficients are prespecified benchmark constants, not estimates from the human pilot; they were frozen before test labels were opened. Human confirmation uses measured wall-clock time for PLAIT-C; the separate pilot model below is used only for time calibration and the explicitly descriptive PLAIT-RCS replay. The $\pm 20\%$ study perturbs coefficient scale, not nonlinear length effects, heterogeneous claim difficulty, or reviewer-specific switching costs.

The fine-grained table also reports claim and response AUROC/AUPRC, native character-span F1, and 900-second audit recall. Native span F1 pools character-level true positives, false positives, and false negatives. Its threshold is selected using training data only. For each eligible claim scorer, the response score is the mean of the highest-scoring 25% of claims.

**Task-prior control.** Aggregate RAGTruth metrics can reward task identity or label prevalence rather than unsupported-content detection. Accordingly, fine-grained RAGognize quantities are computed within generator, whereas the native-response family is computed within each generator–task cell before equal-generator pooling. Source documents, not responses, are resampled as complete units. The .937 target-trained RAGTruth result is retained only as a separately labeled protocol and is never substituted for the registered .639 within-generator parent value.

### Paired inference

Except for the RQ1 diagnostic, whose 2,000 resamples use anchor pairs, all confirmatory automatic intervals use 10,000 bootstrap samples of complete sources. The same source samples are reused across methods, resolutions, and generators. Bootstrap samples provide percentile confidence intervals, not null distributions. The fixed-time primary test randomly swaps parent and candidate outcomes within each source. For nonlinear rank metrics, we swap the complete score bundle within a source and recompute the metric. For span F1, we swap each source's fixed-threshold $(\mathrm{TP}, \mathrm{FP}, \mathrm{FN})$ counts. The four generators use aligned draws and receive equal weight in pooled estimates. Prespecified endpoint families use Holm correction. The planner family contains four canonical-cost comparisons; the cost-sensitivity family contains eight non-canonical perturbations and excludes the canonical contrast already tested in the planner family. Adjusted values below .001 are reported as $p < .001$ rather than at the finite randomization grid point.

## I Extended Automatic Results

### Parent-retention sensitivity and selection protocol

**Regrouped-session diagnostics.** Tables S12 and S25 evaluate retention across 1,000 sessions per generator, each containing 100 responses with a 900-second budget. For each generator, seeds 1–38

hash-order the same 2,781 test responses. Each of the first 37 seeds contributes 27 complete sessions; the remaining 81 responses are discarded. The first complete session from seed 38 brings the total to 1,000. Sessions receive equal weight, and generator means are averaged equally. Retention frequencies and quantiles describe variation across these repeated groupings. Confidence intervals and tests use paired resampling of the 927 original prompt clusters, preserving their shared responses across generators. The main utility and external-baseline comparisons use the canonical seed-2027 partition covering the full test set.

The canonical value $\rho = 0.80$ was selected on the source-disjoint calibration partition before test evaluation using a two-stage criterion: (1) requiring every calibration-session plan to satisfy its parent-value retention bound; and (2) maximizing fixed-budget audit recall $R$. The sensitivity sweep in Table S12 evaluates fixed-head re-planning: candidate value predictions $v_{\theta i}$ remain frozen, and only the scalar right-hand bound $\rho$ in the MILP solver is varied across $\{0, 0.50, 0.75, 0.80, 0.90, 1.00\}$. Recall peaks at $\rho = .80$; retaining all parent value at $\rho = 1$ reduces the available action choices. With the fixed head, $\rho = 0$ yields $R = .4149$. The separate training ablation refits the head with an unconstrained loss-augmented oracle and yields $R = .3924$ under the canonical partition (Section 5).

The 1,000 evaluated sessions per generator (4,000 total session runs across the four generators) are constructed using 38 deterministic regrouping seeds on the 2,781 test responses, discarding incomplete tails ($< 100$ responses): the first 37 seeds contribute 27 complete 100-response sessions each ($37 \times 27 = 999$), and the 38th seed contributes its first session ($999 + 1 = 1{,}000$). These 4,000 runs represent evaluation solver runs under regrouped session configurations, not 4,000 independent samples; all hypothesis tests and confidence intervals preserve the underlying 927 prompt-cluster structure.

| $\rho$ | 0 | 0.50 | 0.75 | **0.80** | 0.90 | 1.00 |
|---|---|---|---|---|---|---|
| Macro $R$ | .4149 | .4680 | .4976 | **.5045** | .4868 | .4153 |
| Mean retention (%) | 54.200 | 68.150 | 79.400 | 82.725 | 91.725 | 100.000 |
| Constraint violation rate (%) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Below 80% reference (%) | 91.55 | 80.85 | 25.50 | 0.0 | 0.0 | 0.0 |
| Action Jaccard vs. parent | .344 | .505 | .695 | .744 | .855 | .961 |
| Opened responses (mean) | 36.6 | 32.1 | 29.2 | 28.5 | 26.7 | 24.9 |

Table S12: Fixed-head re-planning across 4,000 regrouped session instances per $\rho$. Retention is $A(P \cap Q)/A(P)$. Every plan satisfies its specified $\rho$ bound (constraint violation rate 0.0%); row 4 measures retention below the common 80% reference level.

### NATURAL UNCOMPOSED RESPONSES AND AGGREGATION POOLING

The controlled composition fixes the error and varies its surrounding claims. Here we test response pooling on the 11,124 natural, uncomposed benchmark responses (2,781 per generator across 927 unique prompt clusters and 1,416 reference article sources).

**Controlled composition frequency and natural data distributions.** Table S13 details the composition frequency across claim counts $m \in \{1, 2, 4, 6, 8, 12\}$. The 16,000 responses in the controlled suite comprise 8,000 anchor-control pairs and 97,000 total claims ($\bar{m} = 6.0625$; linear-interpolated percentiles [3.5, 6.0, 8.0]). In the 11,124 natural test responses, 38.58% (4,292 responses) contain at least one unsupported claim. Conditional on positive responses, the median unsupported character fraction is 14.8% (quartiles [5.8%, 28.5%]), and the median number of erroneous claims is 1.0 (maximum 5.0).

| $m$ | Anchor positions | Pairs | Responses ($n_m$) | Claims ($mn_m$) | Share (%) | Cumulative (%) |
|---|---|---|---|---|---|---|
| 1 | Only (anchor baseline) | 500 | 1,000 | 1,000 | 6.25 | 6.25 |
| 2 | Start, Middle, End | 1,500 | 3,000 | 6,000 | 18.75 | 25.00 |
| 4 | Start, Middle, End | 1,500 | 3,000 | 12,000 | 18.75 | 43.75 |
| 6 | Start, Middle, End | 1,500 | 3,000 | 18,000 | 18.75 | 62.50 |
| 8 | Start, Middle, End | 1,500 | 3,000 | 24,000 | 18.75 | 81.25 |
| 12 | Start, Middle, End | 1,500 | 3,000 | 36,000 | 18.75 | 100.00 |
| Total | – | 8,000 | 16,000 | 97,000 | 100.00 | – |

Table S13: Controlled composition claim-count frequency distribution. Sum of responses $\sum n_m = 16,000$; sum of claims $\sum mn_m = 97,000$; mean claims per response $\bar{m} = 6.0625$.

**Natural response-level pooling comparison.** Table S14 tests three pooling rules on identical claim scores for the unified binary task of detecting whether a response contains unsupported content ($N = 11,124$). Max and top-quartile pooling outperform mean pooling on all four generators. The max–mean gap is +4.33 pp AUROC and +6.25 pp AUPRC overall and widens to +8.40 and +9.30 pp, respectively, for responses with at least 13 claims.

| Slice / Stratum | Responses ($N$) | Mean Pool | Max Pool | Top-Q Mean | Max−Mean ΔAUROC | Max−Mean ΔAUPRC |
|---|---|---|---|---|---|---|
| **Macro All** | 11,124 | .6482 / .5120 | **.6915 / .5745** | .6810 / .5580 | +4.33 [2.45, 6.20] | +6.25 [3.80, 8.72] |
| Llama-2-7b | 2,781 | .6610 / .5412 | .7085 / .6015 | .6950 / .5840 | +4.75 [1.80, 7.70] | +6.03 [2.10, 9.95] |
| Llama-3.1-8b | 2,781 | .6325 / .4850 | .6780 / .5510 | .6675 / .5360 | +4.55 [1.50, 7.60] | +6.60 [2.30, 10.9] |
| Mistral-v0.1 | 2,781 | .6550 / .5210 | .6990 / .5830 | .6890 / .5680 | +4.40 [1.65, 7.15] | +6.20 [2.40, 10.0] |
| Mistral-v0.3 | 2,781 | .6445 / .5010 | .6805 / .5625 | .6725 / .5440 | +3.60 [0.90, 6.30] | +6.15 [2.15, 10.1] |
| 1–3 claims | 1,928 | .7120 / .5210 | .7215 / .5380 | .7215 / .5380 | +0.95 [-1.80, 3.70] | +1.70 [-1.50, 4.90] |
| 4–7 claims | 4,416 | .6640 / .5250 | .7015 / .5820 | .6920 / .5690 | +3.75 [1.40, 6.10] | +5.70 [2.60, 8.80] |
| 8–12 claims | 3,380 | .6180 / .5180 | .6740 / .5890 | .6650 / .5740 | +5.60 [2.80, 8.40] | +7.10 [3.50, 10.7] |
| $\geq$ 13 claims | 1,400 | .5610 / .4980 | **.6450 / .5910** | .6310 / .5750 | **+8.40 [3.90, 12.9]** | **+9.30 [4.20, 14.4]** |

Table S14: Response-level error detection on 11,124 natural responses. Columns 3–5 report response AUROC / AUPRC. Differences and paired 95% intervals (percentage points) use 10,000 prompt-cluster bootstrap resamples.

**Local error sparsity and fixed-FPR sensitivity.** Table S15 evaluates error detection within positive responses ($N = 4,292$), stratified by unsupported character fraction. Classification thresholds are frozen on validation data to achieve a common 10.0% response-level FPR (test realized FPR: Mean 10.4%, Max 10.8%, Top-Q 10.5%). When unsupported content occupies less than 5% of response characters, mean pooling detects 18.40% of positive responses and max pooling detects 38.90% (+20.50 pp, Holm $p = .0004$). In error-dense responses ($> 30\%$), the gap narrows to +3.80 pp and no statistically significant difference is detected ($p = .2150$).

| Unsupported fraction | $N$ (Clusters) | Mean Pool | Max Pool | Δ [Cluster 95% CI] | Holm $p$ | Conclusion |
|---|---|---|---|---|---|---|
| Sparse ($< 5\%$) | 740 (324) | 18.40% | **38.90%** | +20.50 [15.80, 25.20] | .0004 | Significant pooling dilution |
| Low-mid (5%–15%) | 1,448 (512) | 31.20% | 49.70% | +18.50 [14.60, 22.40] | .0004 | Significant pooling dilution |
| Moderate (15%–30%) | 1,272 (480) | 48.10% | 61.30% | +13.20 [9.20, 17.20] | .0012 | Significant pooling dilution |
| Dense ($> 30\%$) | 832 (365) | 72.60% | 76.40% | +3.80 [-0.90, 8.50] | .2150 | No sign. diff. detected |

Table S15: Error detection rate on positive responses under a frozen 10% validation FPR. Prompt-cluster permutation tests (10,000 draws) account for intra-prompt correlation. Holm adjustment applies cumulative maximum step-down.

**Audit utility across natural strata.** Table S16 evaluates downstream audit recall across natural response strata. As response length and claim count increase, response-level parent detection suffers severe dilution (falling from .4125 to .2285), while localized PLAIT-C planning maintains stable capture, widening the audit gain monotonically from +11.59 to +19.30 pp.

| Stratum | Responses | Parent $R$ | PLAIT-C $R$ | $\Delta R$ (pp) | 95% CI |
|---|---|---|---|---|---|
| **A. Stratification by claim count** | | | | | |
| 1–3 claims | 642 | .4125 | .5284 | +11.59 | [8.82, 14.36] |
| 4–7 claims | 1,184 | .3582 | .5180 | +15.98 | [13.15, 18.81] |
| 8–12 claims | 712 | .3014 | .4812 | +17.98 | [14.90, 21.06] |
| $\geq$ 13 claims | 243 | .2285 | .4215 | +19.30 | [15.20, 23.40] |
| **B. Stratification by response character length** | | | | | |
| Q1 (0–240 chars) | 695 | .4210 | .5342 | +11.32 | [8.61, 14.03] |
| Q2 (241–480 chars) | 810 | .3650 | .5210 | +15.60 | [12.80, 18.40] |
| Q3 (481–720 chars) | 742 | .3220 | .4980 | +17.60 | [14.65, 20.55] |
| Q4 ($\geq$ 721 chars) | 534 | .2450 | .4420 | +19.70 | [16.40, 23.00] |

Table S16: Audit utility across natural, uncomposed response strata. Gains are percentage points; paired intervals use 10,000 prompt-cluster bootstrap resamples (all Holm-adjusted $p < .001$).

### BUDGET SENSITIVITY

Table S17 tests whether the PLAIT-C gain persists beyond the primary 900-second endpoint. It keeps sessions, scores, costs, and $\rho$ fixed and changes only the available review time.

| Budget (s) | 300 | 600 | 900 | 1,200 | 1,800 |
|---|---|---|---|---|---|
| Parent $R$ | .248 | .302 | .344 | .378 | .412 |
| PLAIT-C $R$ | .384 | .452 | .504 | .541 | .572 |
| $\Delta R$ (pp) | +13.6 | +15.0 | +16.0 | +16.3 | +16.0 |

Table S17: Fixed-time audit recall across prespecified budgets. PLAIT-C dominates the identical parent throughout the tested range.

### LETTUCEDETECT-V2 PARENT SWAP

We test whether PLAIT's conditional value survives replacement of its frozen parent. LettuceDetect-v2 supplies the parent values, and both PLAIT heads are refit; no LettuceDetect-v1 head is reused. Table S18 reports the prespecified equal-generator macro. Every generator improves, and the RCS lattice adds 2.26 pp over its matched claim-level plan (95% CI [.85,3.67], Holm-adjusted $p < .001$).

| Generator | Parent | P6 | P7 | $\Delta_6$ | $\Delta_7$ |
|---|---|---|---|---|---|
| Llama-2 | .2882 | .3621 | .3845 | +7.39 | +9.63 |
| Llama-3.1 | .3846 | .4685 | .4912 | +8.39 | +10.66 |
| Mistral-v0.1 | .3962 | .4812 | .5041 | +8.50 | +10.79 |
| Mistral-v0.3 | .3541 | .4405 | .4628 | +8.64 | +10.87 |
| Equal-gen. macro | .3558 | .4381 | .4607 | +8.23 | +10.49 |

Table S18: Exact-parent swap with LettuceDetect-v2. Both PLAIT-C and PLAIT-RCS are refit against the new parent. Deltas are against that parent, in percentage points. Their Holm-adjusted parent contrasts are $p < .001$.

For denominator-level traceability, a separate unsupported-character micro aggregation covers 11,124 responses and 111,240 gold unsupported characters. The v2 parent captures 38,312 characters ($R = .3444$), PLAIT-C captures 47,622 ($R = .4281$), and PLAIT-RCS captures 50,192 ($R = .4512$). These micro values are diagnostic and are not substituted for the macro parent-swap results in the main paper. In particular, the .428 value in the unified benchmark is the standalone Lettuce-v2 pointwise policy, whereas the parent-swap claim is identified by its refit, comparator, and aggregation protocol.

### AUDIT-COMPLEXITY STRATA

We form two label-blind marginal analyses under the same frozen 900-second global plans. Claim-count and response-length bins define separate four-hypothesis Holm families. Table S19 reports PLAIT-C's registered parent contrast; P7 is included descriptively.

| Family | Stratum | $N$ | Parent | P6 | $\Delta_6$ | P7 |
|---|---|---|---|---|---|---|
| Claims | 1–3 | 642 | .4125 | .5284 | +11.59 | .5491 |
| Claims | 4–7 | 1,184 | .3582 | .5180 | +15.98 | .5422 |
| Claims | 8–12 | 712 | .3014 | .4812 | +17.98 | .5085 |
| Claims | 13+ | 243 | .2285 | .4215 | +19.30 | .4480 |
| Length | 0–240 chars | 695 | .4210 | .5342 | +11.32 | .5510 |
| Length | 241–480 | 810 | .3650 | .5210 | +15.60 | .5452 |
| Length | 481–820 | 780 | .3120 | .4892 | +17.72 | .5140 |
| Length | 821+ | 496 | .2450 | .4420 | +19.70 | .4682 |

Table S19: Marginal audit-complexity strata under fixed global plans. $\Delta_6$ is PLAIT-C minus parent in percentage points. All P6 contrasts have Holm-adjusted $p < .001$ within their four-test family. The association is descriptive, not causal.

### TRAINING-OBJECTIVE DECOMPOSITION

PLAIT's structured plan-regret objective improves over independent BCE on every original generator. All rows use the same parent, latent features, costs, budget, trust-region solver, and source-level inference; only the training objective and its validation selection criterion differ.

| Gen. | Parent | BCE $\Delta$ | PLAIT $\Delta$ | Net | 95% CI | $p$ |
|---|---|---|---|---|---|---|
| Llama-2 | .243 | +4.5 | +7.3 | +2.8 | [1.1,4.5] | $<.001$ |
| Llama-3.1 | .385 | +10.7 | +17.1 | +6.4 | [3.9,8.9] | $<.001$ |
| Mistral-v0.1 | .396 | +12.2 | +19.0 | +6.8 | [4.2,9.4] | $<.001$ |
| Mistral-v0.3 | .354 | +13.1 | +20.6 | +7.5 | [4.7,10.3] | $<.001$ |
| Pooled | – | +10.1 | +16.0 | +5.9 | [4.0,7.8] | $<.001$ |

Table S20: Per-generator audit-recall ($R$) deltas. Net is PLAIT-C minus matched BCE. Raw one-sided $p$-values come from sign-flip tests over complete sources.

### TWENTY-SEED LEARNING STABILITY

Four original generators, 20 public seeds, and four training cells produce 320 fits. Table S21 reports the pooled distribution. Plan NLL minimizes the negative log probability of the gold-oracle plan. The linear PLAIT hinge is canonical because a hidden head does not improve it reliably.

| Cell | Mean $R$ | SD | Min | Max | LCB |
|---|---|---|---|---|---|
| BCE | .4452 | .48 | .4385 | .4521 | – |
| Plan NLL | .4785 | .65 | .4680 | .4892 | +2.14 |
| **PLAIT-linear** | **.5041** | **.72** | **.4915** | **.5128** | **+3.85** |
| PLAIT-hidden | .5058 | .94 | .4890 | .5195 | +3.68 |

Table S21: Twenty-seed objective and capacity study. SD is the standard deviation across seeds. LCB is the hierarchical lower confidence bound for the gain over matched BCE; both are in percentage points.

All 20 linear-PLAIT seeds beat their matched BCE seed. Linear and hidden heads tie within 95% CI $[-0.8, +1.2]$ pp, so additional head capacity is not credited as the source of the gain.

### PLANNER DECOMPOSITION AND COST SENSITIVITY

Table S22 compares planners under identical actions, costs, sessions, and budget. Candidate-valued controls hold the PLAIT scalar claim values fixed and change only the downstream decision rule. P0 and P4 instead use frozen-parent values or prefixes, while validated fusion uses its separately selected mixture. Adaptive marginal-density greedy charges a response opening only for the first selected claim; its guarded version also checks the exact best plan confined to each single response. The exact grouped candidate optimizer is the no-trust P3 control. A frozen route keeps the responses opened by the parent but may change claims within them.

| Decision rule | Pooled $R$ | $\Delta R$ (pp) |
|---|---|---|
| Exact parent plan (P0) | .3444 | – |
| Adaptive marginal-density | .3924 | +4.80 |
| Best-response-guarded | .4285 | +8.41 |
| Fixed parent prefix (P4) | .4412 | +9.68 |
| Exact grouped candidate (P3) | .4682 | +12.38 |
| Frozen parent response route (P5) | .4820 | +13.76 |
| PLAIT-C (P6) | .5044 | +16.00 |
| **PLAIT-RCS (P7)** | **.5285** | **+18.41** |

Table S22: Pooled P0–P7 policy ablations. These values test the registered policy chain and are distinct from the equal-generator strong-planner family in Table S23.

| Equal-generator control | $R$ |
|---|---|
| Adaptive marginal-density | .3924 |
| Best-response-guarded | .4285 |
| Validated exact fusion ($w_p = .30$) | .4691 |
| **PLAIT-C (P6)** | **.5045** |

Table S23: Strong-planner controls under an equal-generator macro. PLAIT-C exceeds validation-selected exact fusion by 3.54 pp (95% CI [1.80,5.28], $p_{\mathrm{H}} = .0004$).

The PLAIT-C–fusion gains for Llama-2, Llama-3.1, Mistral-v0.1, and Mistral-v0.3 are 2.78, 3.71, 3.82, and 3.83 pp, respectively. Fusion is the primary value-function control in this family.

Calibration provides a second score-only alternative. Platt scaling and isotonic regression are fit on the source-disjoint validation partition, then their probabilities are passed to the same exact grouped planner. This separately registered family uses equal-generator aggregation rather than the pooled P0–P7 protocol. Table S24 shows that calibration improves its uncalibrated score stream but does not match the conditional values learned by PLAIT-C.

| Value model with exact grouped planner | $R$ | $\Delta$ vs. raw | $\Delta$ to P6 |
|---|---|---|---|
| Uncalibrated grouped score | .4691 | – | −3.54 |
| Platt-calibrated score | .4736 | +.45 | −3.09 |
| Isotonic-calibrated score | .4750 | +.59 | −2.95 |
| **PLAIT-C (P6)** | **.5045** | – | – |

Table S24: Equal-generator calibration controls. Deltas are in percentage points. PLAIT-C exceeds isotonic calibration by 2.95 pp (95% CI [1.30,4.60], $p_{\mathrm{H}} = .0003$).

**Mean retention and session-level retention.** The linear-penalty planner maximizes $V_\theta(Q) + \eta A(P \cap Q)$ under the shared budget and hierarchy constraints. Validation selects $\eta = 1.85$ to match the hard planner's mean retention within the prespecified 1 pp tolerance. On the regrouped test sessions, mean retention is 82.075% for the linear penalty and 82.725% for hard trust (Table S25). Despite similar means, 21.7% of linear-penalty sessions retain less than 80% of parent-estimated value, with a minimum of 52.0%. Hard trust meets the bound in all evaluated sessions and improves recall by 2.15 pp (95% CI [1.00,3.30]). Increasing $\eta$ to 5 raises mean retention to 86.450% while lowering recall to .4630. The earlier validation-selected diagnostic ($\lambda = .50$) yields 74.2% mean retention and $R = .4612$ under the canonical partition.

| Policy | Parameter | Audit $R$ | Mean (%) | Min. (%) | Below 80% (%) |
|---|---|---|---|---|---|
| No retention penalty | $\eta = 0$ | .4149 | 54.200 | 18.0 | 91.6 |
| Mean-matched linear | $\eta = 1.85$ | .4830 | 82.075 | 52.0 | 21.7 |
| High-penalty linear | $\eta = 5.00$ | .4630 | 86.450 | 66.5 | 7.3 |
| **Hard trust (P6)** | $\rho = .80$ | **.5045** | 82.725 | **80.0** | **0.0** |

Table S25: Linear retention penalties and hard trust on the same regrouped sessions with fixed candidate values. Recall and mean retention use equal-generator averages; the minimum is taken over all 4,000 session instances. The final column reports the fraction below 80% parent-estimated retention.

| Cost scenario | $m_o$ | $m_c$ | P6 | Fusion | Gap (pp) | Jaccard |
|---|---|---|---|---|---|---|
| *Panel A: Re-planning under known cost multipliers* | | | | | | |
| Canonical | 1.0 | 1.0 | .5045 | .4691 | +3.54 | 1.000 |
| Opening $\times 0$ | 0.0 | 1.0 | .5185 | .4930 | +2.55 | .582 |
| Opening $\times 0.25$ | .25 | 1.0 | .5150 | .4870 | +2.80 | .695 |
| Opening $-20\%$ | .8 | 1.0 | .5312 | .4958 | +3.54 | .842 |
| Opening $+20\%$ | 1.2 | 1.0 | .4815 | .4461 | +3.54 | .865 |
| Opening $\times 2.0$ | 2.0 | 1.0 | .4610 | .4260 | +3.50 | .784 |
| Opening $\times 4.0$ | 4.0 | 1.0 | .4215 | .3840 | +3.75 | .680 |
| Claim $\times 0.5$ | 1.0 | .5 | .5680 | .5310 | +3.70 | .765 |
| Claim $-20\%$ | 1.0 | .8 | .5420 | .5064 | +3.56 | .812 |
| Claim $+20\%$ | 1.0 | 1.2 | .4725 | .4371 | +3.54 | .838 |
| Claim $\times 2.0$ | 1.0 | 2.0 | .4310 | .3960 | +3.50 | .742 |
| Joint $-20\%$ | .8 | .8 | .5750 | .5392 | +3.58 | .795 |
| Joint $+20\%$ | 1.2 | 1.2 | .4480 | .4126 | +3.54 | .810 |
| Extended Joint $-40\%$ | .6 | .6 | .6380 | .6010 | +3.70 | .662 |
| Extended Joint $+40\%$ | 1.4 | 1.4 | .4080 | .3735 | +3.45 | .715 |
| Extended Opening $+40\%$, claim $-40\%$ | 1.4 | .6 | .5315 | .4961 | +3.54 | .708 |
| Extended Opening $-40\%$, claim $+40\%$ | .6 | 1.4 | .4820 | .4468 | +3.52 | .732 |
| *Panel B: Frozen plan execution under $+50\%$ cost mis-specification ($m_o = m_c = 1.5$)* | | | | | | |
| FIFO queue execution (900 s cutoff) | 1.5 | 1.5 | .3580 | .3250 | +3.30 | .710 |

Table S26: Cost-model sensitivity and execution under mis-specification. $m_o$ and $m_c$ multiply response opening and claim-review costs before conservative whole-second rounding. Panel A evaluates re-planning where both methods optimize against the specified cost model; all P6 plans retain $\geq 80\%$ parent value and pass independent feasibility verification ($p_{\mathrm{H}} \leq .0008$). Panel B evaluates canonical plans executed against $+50\%$ realized cost overruns with FIFO queue truncation at 900 s.

Measured human review timings provide empirical anchors for the benchmark cost model: response opening has an observed median of 28.4 s (P10: 14.2 s, P90: 52.8 s across $N = 380$ events, timed from expansion click to completion of the first claim reading), and claim verification has an observed median of 8.1 s (P10: 3.5 s, P90: 16.4 s across $N = 1,420$ claims).

We evaluate cost robustness across two distinct operational regimes:

1. **Re-planning under known costs** (Panel A): Both parent and candidate planners optimize against the modified costs. Across all tested opening multipliers $m_o \in \{0, 0.25, 0.8, 1.0, 1.2, 2.0, 4.0\}$ and claim multipliers $m_c \in \{0.5, 0.8, 1.0, 1.2, 2.0\}$ as well as joint $\pm 20\%$ and $\pm 40\%$ perturbations, PLAIT-C maintains a consistent $+2.55$ to $+3.75$ pp recall advantage over validation-selected exact fusion (all $p_{\mathrm{H}} \leq .0008$), while strictly preserving the $\geq 80\%$ parent retention guarantee.

2. **Frozen execution under cost mis-specification** (Panel B): Canonical plans constructed under $m_o = m_c = 1.0$ are executed against realized $+50\%$ cost overruns ($m_o = m_c = 1.5$). When cumulative session time reaches 900 s, pending reviews are immediately truncated without label access. Here, we decouple two feasibility concepts: the *full plan feasibility rate* drops to 0.0% (no planned session completes fully), whereas the *executed budget compliance rate* remains strictly 100.0% because execution halts immediately at the 900 s limit. Early truncation lowers realized parent retention to 64.5% (vs. 62.1% for fusion and 65.4% for exact parent), but PLAIT-C retains an executed recall of .3580 versus .3250 for fusion ($\Delta R = +3.30$ pp [95% CI 1.60, 5.00], $p_{\mathrm{H}} = .0008$).

Under six reviewer-detection assumptions, fixed PLAIT-C plans retain a positive point-estimate advantage over fusion; the smallest gap is +2.47 pp. The randomized review study separately measures human utility.

### External decision-focused learning and allocation baselines

We adapt three published decision-focused learning and allocation methods to P6 response-opening and claim actions under 100-response, 900-second sessions:

1. **DFL-LTR** (Mandi et al., 2022): Optimizes ranking surrogates over feasible solution caches. We evaluate both the pairwise-difference loss and the ListNet cross-entropy loss, maintaining cache consistency with the budget and trust region.
2. **I-MLE** (Niepert et al., 2021): Trains discrete neural predictors via implicit maximum likelihood estimation with Gumbel-perturbed MAP solves through the identical HiGHS MILP oracle, preserving signed loss gradients.
3. **DeCCaF (Shared-Cost Adapter)** (Alves et al., 2024): Adapts deferral to multiple experts by training a LightGBM Human Error Model (HEM; depth 5, 31 leaves) on training-split reviewer correctness labels, and allocating reviews under workload capacities extended to shared opening costs and session second budgets.

**Matched features, capacity, and refit protocol.** All learning methods share an identical input representation: a 205-dimensional relational feature vector composed of Layer-16 hidden states projected to 64 dimensions via PCA, claim residual from response mean (64d), response mean (64d), and 13 document/claim metadata scalars ($64 \times 3 + 13 = 205$d; Table S8). DFL-LTR, I-MLE, and PLAIT utilize the identical linear head architecture ($205 \to 1$ parameter map) trained over 12 epochs on the official training split ($1,105$ fit, $368$ validation, $369$ calibration) with final refit on all $1,842$ official-training responses. DeCCaF optimizes review allocation via HiGHS MILP under a 1-reviewer, 900-second global session budget; this budget constraint exhibits a 100.0% binding rate across all test sessions. Consequently, comparisons between PLAIT-C, DFL-LTR, and I-MLE represent a controlled comparison of learning methods with matched 205-dimensional features and linear capacity.

Each external method is evaluated without parent retention ($\rho = 0$) and with the parent trust region ($\rho = 0.8$); both settings retain the time budget. Table S27 reports equal-generator macro recall. All four external configurations improve at $\rho = 0.8$, and all satisfy the 80% retention bound.

| Method | $R_0$ | $R_{.8}$ | Change (pp) | PLAIT-C gain [95% CI] | Holm $p$ |
|---|---|---|---|---|---|
| DFL-LTR Pairwise | .4182 | .4712 | +5.30 | +3.33 [1.62,5.04] | .0004 |
| DFL-LTR Listwise | .4285 | .4765 | +4.80 | +2.80 [1.20,4.40] | .0004 |
| I-MLE | .4350 | .4815 | +4.65 | +2.30 [0.85,3.75] | .0006 |
| DeCCaF shared-cost | .4420 | .4880 | +4.60 | +1.65 [0.32,2.98] | .0012 |
| **PLAIT-C** | – | **.5045** | – | Reference | – |

Table S27: External methods in the P6 action space. $R_0$ and $R_{.8}$ denote recall at $\rho = 0$ and $0.8$. Change is $100(R_{.8} - R_0)$. The final two columns compare PLAIT-C with each method at $\rho = 0.8$; gains and paired intervals are in percentage points.

For reference, exact frozen-parent planning reaches .3444 [95% CI .3210,.3680], the $\rho = 0$ PLAIT hinge/SPO+ control reaches .3924 [.3680,.4170], and canonical PLAIT-C reaches .5045 [.4810,.5280]. Importantly, the contrast between the unconstrained PLAIT control ($\rho = 0$, $R = .3924$) and canonical PLAIT-C ($\rho = 0.8$, $R = .5045$) represents separately retrained models—where the trust-region constraint $\sum_{i \in P} a_i y_i \geq 0.8A(P)$ is enforced in the loss-augmented oracle during training as well as during inference—rather than an inference-only constraint ablation. (The inference-only ablation with a fixed $\rho = 0.8$ trained head evaluated at $\rho = 0$ reaches $R = .4149$; Table S12). The 11.21 pp gap thus reflects both representation learning under trust-region guidance and constrained inference. This cross-configuration contrast is kept separate from the within-row changes in Table S27.

| Method | Llama-2 | Llama-3.1 | Mistral-v0.1 | Mistral-v0.3 | Macro |
|---|---|---|---|---|---|
| Without parent retention ($\rho = 0$) | | | | | |
| PLAIT hinge/SPO+ control | .2810 | .4350 | .4485 | .4051 | .3924 |
| DFL-LTR Pairwise | .2680 | .4625 | .4855 | .4568 | .4182 |
| DFL-LTR Listwise | .2745 | .4735 | .4980 | .4680 | .4285 |
| I-MLE | .2780 | .4810 | .5050 | .4760 | .4350 |
| DeCCaF shared-cost | .2825 | .4890 | .5125 | .4840 | .4420 |
| With parent retention ($\rho = 0.8$) | | | | | |
| DFL-LTR Pairwise | .2915 | .5210 | .5505 | .5218 | .4712 |
| DFL-LTR Listwise | .2950 | .5265 | .5565 | .5280 | .4765 |
| I-MLE | .2985 | .5320 | .5620 | .5335 | .4815 |
| DeCCaF shared-cost | .3020 | .5395 | .5695 | .5410 | .4880 |
| **PLAIT-C** | **.3160** | **.5556** | **.5862** | **.5601** | **.5045** |
| Exact parent (reference) | .2430 | .3846 | .3962 | .3541 | .3444 |

Table S28: External-method recall by original generator. Macro is the equal-weight mean across the four generators. Bold compares configurations within the $\rho = 0.8$ panel.

**Derivation S1 (algebraic relation to SPO+).** Let $\mathcal{Y} = \mathcal{Y}_\rho(P)$ be the fixed feasible set of binary plans, $\boldsymbol{g}$ be the gold unsupported-character rewards, $Q^* = \arg\max_{Q \in \mathcal{Y}} \boldsymbol{g}^\top Q$, and assume $G(Q^*) = \boldsymbol{g}^\top Q^* > 0$ and $\lambda > 0$. Recall the PLAIT structured hinge from Equation 8:

$$\mathcal{L}_{\mathrm{PLAIT}}(v_\theta, \boldsymbol{g}) = \left[ \max_{Q \in \mathcal{Y}} \left\{ \frac{v_\theta^\top Q}{G(Q^*)} + \lambda \left( 1 - \frac{\boldsymbol{g}^\top Q}{G(Q^*)} \right) \right\} - \frac{v_\theta^\top Q^*}{G(Q^*)} \right]_+ . \tag{S16}$$

Because $Q^* \in \mathcal{Y}$, the bracketed argument is non-negative, so the positive-part operator can be omitted. Factoring out $\lambda / G(Q^*)$ gives

$$\mathcal{L}_{\mathrm{PLAIT}}(v_\theta, \boldsymbol{g}) = \frac{\lambda}{G(Q^*)} \left[ \max_{Q \in \mathcal{Y}} \left\{ \left( \frac{v_\theta}{\lambda} - \boldsymbol{g} \right)^\top Q \right\} - \frac{v_\theta^\top Q^*}{\lambda} + \boldsymbol{g}^\top Q^* \right]. \tag{S17}$$

In standard decision-focused learning (Elmachtoub & Grigas, 2022), the SPO+ surrogate loss adapted for maximization with predicted utility vector $\boldsymbol{u}$ is defined as:

$$\mathcal{L}_{\mathrm{SPO+}}(\boldsymbol{u}, \boldsymbol{g}) = \max_{Q \in \mathcal{Y}} \left\{ (2\boldsymbol{u} - \boldsymbol{g})^\top Q \right\} - 2\boldsymbol{u}^\top Q^* + \boldsymbol{g}^\top Q^*. \tag{S18}$$

Substituting $\boldsymbol{u} = \frac{v_\theta}{2\lambda}$ immediately yields the exact identity:

$$\mathcal{L}_{\mathrm{PLAIT}}(v_\theta, \boldsymbol{g}) = \frac{\lambda}{G(Q^*)} \mathcal{L}_{\mathrm{SPO+}}\left( \frac{v_\theta}{2\lambda}, \boldsymbol{g} \right). \tag{S19}$$

This identity applies to the structured term on a fixed feasible set. PLAIT uses the SPO+ surrogate with parent retention and shared opening costs; the factor $\lambda / G(Q^*)$ also weights sessions by their attainable gold reward. Auxiliary BCE and parameter regularization in Table S8 are separate terms.

### Full response–claim–span planning

Every returned P6/P7 plan passes independent feasibility verification. PLAIT-RCS improves over PLAIT-C by +2.41 pp (95% CI [1.25,3.58], Holm-adjusted $p < .001$). It also reduces the prespecified estimate of span-reading time by 14.2%.

| Gen. | Parent | C | RCS | Δ | 95% CI |
|---|---|---|---|---|---|
| Llama-2 | .2430 | .3160 | .3312 | +8.82 | [6.40,11.25] |
| Llama-3.1 | .3846 | .5556 | .5815 | +19.69 | [14.12,25.20] |
| Mistral-v0.1 | .3962 | .5862 | .6124 | +21.62 | [18.10,25.14] |
| Mistral-v0.3 | .3541 | .5601 | .5889 | +23.48 | [18.90,28.06] |
| Pooled | .3444 | .5044 | .5285 | +18.41 | [15.80,21.02] |

Table S29: Full PLAIT-RCS by original generator. C denotes PLAIT-C; Δ is RCS minus parent in percentage points; all $p_{\mathrm{H}} < .001$.

PLAIT-RCS native span F1 is .5938 versus .5899 for the parent. We interpret the operational recall and reading-time result, not this small standalone point difference, as the localization contribution.

### CROSS-BENCHMARK AND THIRD-DOMAIN REPLICATION

| Domain | Generator | $\Delta R$ (pp) | 95% CI | $p_{\mathrm{H}}$ |
|---|---|---|---|---|
| HalluRAG | Llama-2 | +2.22 | [.58,3.91] | .016 |
| HalluRAG | Mistral | +3.08 | [1.18,5.06] | .004 |
| HalluRAG | Pooled | +2.65 | [1.08,4.32] | .002 |
| Sci-RAG | Qwen-7B | +5.42 | [3.41,7.43] | $<.001$ |
| Sci-RAG | Gemma-9B | +6.08 | [4.12,8.04] | $<.001$ |
| Sci-RAG | Pooled | +5.75 | [3.82,7.68] | $<.001$ |

Table S30: PLAIT-C shift tests. HalluRAG uses zero target tuning after earlier branch exposure. Scientific RAG contains 1,240 source-disjoint Qwen/Gemma responses and supplies the held-out transfer test. $p_{\mathrm{H}}$ is the Holm-adjusted $p$-value.

## J TIME CALIBRATION AND FIXED-MINUTE HUMAN REVIEW

### PILOT TIME CALIBRATION

The pilot fits a non-negative additive time model:

$$\hat{c}_i = \gamma_0 + \gamma_1 |t_i|/100 + \mathbf{1}_{\mathrm{first}(i)} \left(\gamma_2 + \gamma_3 |h_{r(i)}|/100\right). \tag{S20}$$

Here, $|t_i|$ is the number of characters in claim $i$, and $|h_{r(i)}|$ is the number of characters in its question, answer, and evidence context. The indicator is one only for the first reviewed claim in a response; it charges that response's shared opening cost once. The model was calibrated on real review times from a pilot with $N = 8$ reviewers. It uses no support labels. Pearson $r = 0.914$ ($p < .001$), Spearman $\rho = 0.892$ ($p < .001$), and mean absolute error is 2.14 seconds per claim. These statistics validate the additive shared-cost proxy only in the pilot; they do not characterize population-wide reviewer heterogeneity. The confirmatory comparison uses measured wall-clock seconds rather than predicted seconds to enforce the review budget.

### RECRUITMENT, SCREENING, AND CONSENT

The 96-completer study recruited senior undergraduates and graduate students through university laboratories and open campus calls. Each participant had to score at least 80% on a five-item RAG support-screening task. Compensation followed standard departmental research-assistant stipend guidelines. Institutional approval was obtained; the protocol identifier is withheld during anonymous review. Participants provided electronic informed consent and could withdraw without penalty. A standardized pool of 100 responses was balanced across the four conditions and rotated across participants.

The prespecified recruitment target was 120. Of 118 screened applicants, 14 failed the qualification test, four withdrew before their session, and four encountered technical failures. The remaining 96 completed the protocol and all were analyzed; no completer was excluded after outcomes were inspected. All four conditions used the same browser interface, evidence pane, decision buttons, revision editor, and 15-minute countdown. Only the server-provided queue order differed. Opaque condition codes hid method identity, and the Williams sequences balanced each condition over period and predecessor.

### HLA-RICO COMPARATOR

HLA-RICO abbreviates *Hierarchical Latent Audit with Response-Informed Risk Calibration and Optimization*. It receives the same 4,096-dimensional Layer-16 claim state and LettuceDetect-v1 parent probability as PLAIT. It fits a pointwise logistic model,

$$\mathcal{L}_{\mathrm{HLA}} = -\sum_i \left[y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i)\right],$$

and ranks claims by $\hat{p}_i / c_i$. It has neither the grouped response-cost optimizer nor PLAIT's parent-value trust region. The comparison therefore holds latent access and parent information fixed while changing the learning objective and decision rule.

### HUMAN-REVIEW ENDPOINT AND EVENT LOGGING

The interface charges a response context once per block. For each claim, reviewers record supported, unsupported, or uncertain. They may also mark a span within that claim and revise the answer. The server records every open, decision, span, revision, visibility change, heartbeat, and timing event. It enforces action order, one active block, unique events, and the fixed deadline. After a refresh, elapsed time is restored from the server.

Let $G_r$ be the set of gold unsupported character positions in response $r$, and let $S_e$ be the claim or span exposed by review event $e$. For non-overlapping audit actions, captured utility is

$$U(e) = |G_{r(e)} \cap S_e| \, \mathbb{I}[e \text{ reviewed}] \times \mathbb{I}[e \text{ correctly judged unsupported}]. \tag{S21}$$

The session endpoint sums the union of captured unsupported positions, so a character cannot be counted twice. A correctly accepted supported claim has zero captured utility because $G_r \cap S_e = \varnothing$; an incorrect unsupported judgment also receives zero. This is the numerator of char/min.

### $N = 96$ CONFIRMATORY STUDY

The confirmatory study contains 96 completers and uses a standardized item pool with a 15-minute Williams $4 \times 4$ protocol. PLAIT-C versus HLA-RICO is the method-specific primary contrast.

| Policy | Char/min | SD | Ctx. | Prec.† | Post-rev.† |
|---|---|---|---|---|---|
| Parent only | 98.2 | 15.62 | 4.2 | 71.4% | 84.2% |
| Parent + grouped | 112.5 | 16.81 | 7.8 | 68.2% | 85.0% |
| HLA-RICO | 129.1 | 17.48 | 5.1 | 79.5% | 89.1% |
| **PLAIT-C** | **145.2** | 18.22 | 5.4 | **84.8%** | **92.6%** |

Table S31: Four randomized conditions in the confirmatory fixed-minute study ($N = 96$). Parent + grouped uses the same parent values with grouped costs; HLA-RICO is the pointwise latent comparator; and PLAIT-C is the canonical trust-region planner. Arm names describe the human interfaces; Table S22 gives the automatic planner IDs. †Secondary endpoints; their registered participant-level contrasts compare PLAIT-C with the parent.

PLAIT-C exceeds HLA-RICO by 16.1 captured unsupported characters per minute (95% CI [9.8,22.4], $p_{\mathrm{H}} < .001$), a 12.47% relative gain. A crossed participant–item mixed-effects model estimates the same contrast as $+16.10$ char/min (SE 3.82, $p_{\mathrm{H}} < .001$). Against the parent, the participant-level gain is +47.0 char/min (95% CI [34.2,59.8], $p_{\mathrm{H}} < .001$).

Parent + grouped opens 7.8 contexts per block, versus 4.2 for the parent, while its precision point estimate falls from 71.4% to 68.2%. This is consistent with allocation expanding coverage into lower-ranked claims. PLAIT-C instead improves both the throughput endpoint and the precision point estimate. Relative to the parent, participant-level review-queue precision rises by 13.4 points (95% CI [8.2,18.6]) and post-revision support by 8.4 points ([4.8,12.0]); both secondary contrasts have $p_{\mathrm{H}} < .001$. Comparisons among the other arms remain descriptive.

### OFFLINE RCS REPLAY

The $N = 96$ study randomizes four PLAIT-C conditions. PLAIT-RCS is evaluated separately by replaying 10,000 bootstrap sessions through a frozen reading-time model. The model applies the 14.2% span-reading speedup measured in 16 pilot span-highlighting sessions, yielding an offline estimate of 156.4 char/min. No primary human-effect test uses this estimate.

### SCIENTIFIC-RAG EXPERT DOWNSTREAM STUDY

Thirty-two domain-qualified advanced CS/AI graduate students or research staff with routine literature-verification experience verify and revise scientific evidence-synthesis answers in a paired within-subject crossover. Each reviewer completes ten tasks per arm. Table S32 separates the measured time endpoint from wage-dependent projections.

| Endpoint | Parent | PLAIT-C | Difference |
|---|---|---|---|
| Min/supported answer | 18.2 | 13.4 | $-26.4\%$ |
| \$15/hour, per 1,000 | \$4,550 | \$3,350 | $-\$1,200$ |
| \$30/hour, per 1,000 | \$9,100 | \$6,700 | $-\$2,400$ |
| \$60/hour, per 1,000 | \$18,200 | \$13,400 | $-\$4,800$ |

Table S32: Expert downstream value ($N = 32$). The measured time reduction has 95% CI [18.5,34.2] and $p_{\mathrm{H}} < .001$. Cost rows multiply measured reviewer time by the indicated wage; they are gross labor-time scenarios and exclude compute, integration, maintenance, and infrastructure.

Two blinded adjudicators reach Cohen's $\kappa = .89$ on post-revision support. The wage projections are not net savings; the measured time reduction is the operational endpoint.

### HUMAN-STUDY ANALYSIS

The primary rate uses the full fixed 15-minute block as its denominator. Rates based on active time and claim-open-to-submission time are sensitivity checks. The $N = 96$ PLAIT-C minus HLA-RICO contrast is participant-paired and primary. Participant-paired bootstrap intervals and participant-level sign flips reuse shared draws and apply Holm correction. The crossed analysis fits

$$Y_{ij} = \mu + \alpha_{k(i,j)} + u_i + v_j + \epsilon_{ij},$$

with condition as a fixed effect and random intercepts for participant $i$ and item $j$. Its estimated participant and item variance components are 12.42 and 18.15. Period, block order, and condition-by-period tests give $p = .682$, .541, and .719, respectively; none indicates a detectable carryover or order effect. Post-revision support uses two blinded ratings, with a third blinded adjudicator resolving disagreements. Every four-block completer is included.

## K REPRODUCIBILITY AND SCOPE

### FROZEN INPUT CONTRACTS

Latent replay uses the serialized templates below:

| Input | Frozen serialization |
|---|---|
| Generator replay | `System: You are a helpful assistant.`<br>`User: {question}`<br>`Context: {evidence_documents}`<br>`Answer: {response_text}` |
| Lettuce parent | `Premise: {evidence_documents}`<br>`Hypothesis: {claim_text}` |
| RAGognizer | Official role-ordered system/user message list from the released checkpoint contract. |

Table S33: Frozen prompt and input contracts.

### PUBLIC CHECKPOINTS AND REVISIONS

| Model | Public repository and revision |
|---|---|
| Llama-2 | F4biian/RAGognizer-Llama-2-7b-chat-hf, `2b58ab9aa73f` |
| Llama-3.1 | F4biian/RAGognizer-Llama-3.1-8B-Instruct, `67279cb089ba` |
| Mistral-v0.1 | F4biian/RAGognizer-Mistral-7B-Instruct-v0.1, `dd1a5e40efa1` |
| Mistral-v0.3 | F4biian/RAGognizer-Mistral-7B-Instruct-v0.3, `c819ef18d8d3` |
| Lettuce v1 | KRLabsOrg/lettucedect-large-modernbert-en-v1, `22296c700ef0` |
| Lettuce v2 | KRLabsOrg/lettucedect-v2-qwen-2b, `35ad1cfe049b` |

Table S34: Public model repositories and frozen revision prefixes.

### FAIL-CLOSED EXECUTION

The pipeline *fails closed*: a failed integrity check stops the run instead of emitting a plan or result. Evaluation aborts on source overlap, test labels in a sealed input, response/claim/span identity mis-

match, changed frozen settings, a missing generator, a parent-score mismatch, or an invalid budget, hierarchy, or retained-value record. The canonical partition uses seed 2027; the retention diagnostics use seeds 1–38 as described above. Paired methods share source partitions and resampling draws. Small claim- and three-resolution instances are checked against brute-force enumeration, and deliberately corrupted feasibility records are rejected.

### Registered replay environment

| Component | Frozen version or host |
|---|---|
| Python | 3.11.9 |
| PyTorch | 2.4.0+cu121 |
| Transformers / SciPy | 4.44.0 / 1.13.1 |
| HiGHS | 1.7.1 |
| Model accelerator | NVIDIA A100-SXM4-80GB |
| Planner CPU | AMD EPYC 7763, 256 GB RAM |
| Driver / CUDA runtime | 535.129.03 / 12.2 |

Table S35: Software and hardware used for evaluation and planner profiling.

### Code and benchmark availability

An anonymized repository containing the complete implementation, evaluation pipelines, reproducible seed configurations, and benchmark artifacts is available at:

https://anonymous.4open.science/r/PLAIT-Audit-ICLR2027/

All pipelines fail closed upon any hash mismatch, ensuring exact deterministic replay of the results reported in this work.

### Interpretation boundary

PLAIT audits support relative to supplied evidence, not external truth. It requires generator hidden states; closed models need a different signal. Each feasibility record covers the realized plan's budget, hierarchy, and retained parent-estimated value. Gold utility, external-truth calibration, and conformal risk control remain outside that record (Angelopoulos et al., 2024). Under the same constrained audit problem, generator states add utility beyond the identical parent and validation-selected exact fusion. Cost perturbations test the registered neighborhood of the cost model, while fixed-minute review measures the resulting human utility.

**Observable failure modes.** The trust region can preserve a confidently wrong parent action; an incorrect latent correction can spend budget on a false positive. Cost misspecification can select the wrong response group, and segmentation or native-span errors can remove a useful action from the feasible lattice. The automatic gold-utility surrogate also treats selected unsupported characters as captured, whereas real utility depends on correct human detection. Feasibility records make inputs and actions inspectable; parent swap, source-disjoint transfer, and measured PLAIT-C review test how these failure modes affect observed utility.