PLAIT: From Hallucination Scores to Parent-Preserving Audit Plans

Anonymous submission

Abstract

Hallucination detectors estimate which text is unsupported, but fixed-time review requires a feasible choice of which response, claim, or span to inspect. We identify the *resolution–allocation gap*: response pooling sharply dilutes localized errors, whereas independent fine-grained scores ignore shared context-opening costs. We introduce PLAIT (*Parent-Preserving Latent Auditing with Interdependent Triage*), which turns a frozen detector into a response–claim–span audit plan. Generator hidden states refine the parent's review values. PLAIT trains these refinements against whole-plan regret and selects nested actions with an exact planner. A parent-value trust region limits departures from the parent plan, while an independent verifier recomputes budget, hierarchy, and retained value. Across four original generators, PLAIT-RCS raises 900-second recall by 18.41 points over exact planning with frozen parent values. In a separate equal-generator comparison, PLAIT-C remains 3.54 points above validation-selected exact fusion. Gains relative to the corresponding frozen parent remain positive after parent replacement and source-disjoint transfer. In randomized fixed-minute review, PLAIT-C captures more unsupported content per minute. The staged controls isolate the roles of conditional value, plan-level learning, retention, and action resolution.

1 Introduction

Retrieval-augmented generation grounds answers in retrieved evidence (Lewis et al. 2020), yet failures can remain local: a long response may contain ten supported claims and one unsupported date, attribution, or conclusion. A detector can flag that claim, but a reviewer with limited time cannot inspect every flag. Detection asks *where might support fail?* Auditing asks *which content should be inspected next, at what resolution, and within which budget?* A score answers the first question; deployment requires an answer to the second.

The resolution–allocation gap. A common evaluation shortcut equates a better detection score with a better audit. This equivalence requires two assumptions: the scoring resolution must preserve a local error, and each scored item must be reviewable independently. Both assumptions fail in fixed-time review. First, response pooling causes *resolution dilution*: averaging weakens a localized signal before any audit policy acts. Under a centered residual model, averaging m claims shrinks the mean shift of one unsupported claim to $1/m$, and its excess quadratic signal to $1/m^2$. Natural compositions show the same effect: with one unsupported claim among 12, a response centroid retains 2.0% of its single-claim effect, versus 14.6–15.2% for claim-level signals. A monitor cannot prioritize an error whose identity its resolution has removed.

Preserving the claim is necessary, but not sufficient: pointwise scores still do not define an audit. Suppose opening either of two responses costs 40 s. Response A contains claims $(\text{risk}, \text{cost}) = (.9, 20\,\text{s})$ and $(.4, 30\,\text{s})$; response B contains $(.7, 10\,\text{s})$. Under a 90-second budget, a score-per-second rule opens B first and captures .7 total score. Opening A instead pays its context cost once and captures both claims, for 1.3. No independent claim threshold repairs this failure. The marginal cost of a claim depends on which sibling claims share its context. A scalar score describes one item, but an audit must choose a coupled set. We call this second failure *allocation loss*. Together, resolution dilution and allocation loss create the *resolution–allocation gap*.

PLAIT closes this gap by replacing independent ranking with one constrained decision. A frozen detector first defines the *parent* plan, and exact grouped optimization charges shared opening costs once. PLAIT then uses hidden states recorded from the model that produced the answer to refine the parent's review values. It trains those refinements against plan regret and selects nested response–claim–span (RCS) actions. A parent-value trust region preserves a chosen fraction of the parent plan's estimated value. Inference returns both a review queue and a feasibility record that a separate verifier can recompute. Figure 1 summarizes this decision. **PLAIT** abbreviates *Parent-Preserving Latent Auditing with Interdependent Triage*.

Contributions. We make three contributions. First, we define the resolution–allocation gap and measure resolution dilution on natural unsupported claims. Second, we formulate evidence review as a parent-preserving RCS decision trained against whole-plan regret. Third, same-parent ablations test which design choices contribute to the gain. Parent replace-

ment, source-disjoint transfer, and blinded review then test whether it persists beyond the original detector and benchmark.

2 Related Work

Detection and localization. Sample- and evidence-based monitors score factual consistency (Manakul, Liusie, and Gales 2023; Tang, Laban, and Durrett 2024). RAGTruth supplies natural character-span annotations (Niu et al. 2024); LettuceDetect, RAGognizer, CORTEX, RLSeek, and LAFaCT localize unsupported or fact-critical text (Kovács and Recski 2025; Ridder, Lessel, and Schilling 2026; Furumai et al. 2026; Huang et al. 2026; Wang et al. 2026). ReDeEP decouples copying-head and feed-forward signals; LUMINA measures external-context and internal-knowledge use (Sun et al. 2025; Yeh, Li, and Mallick 2026). Grounded probes can localize errors but remain domain-sensitive (CH-Wang et al. 2024); long-form probes expose efficient response-level factuality signals (Han et al. 2025). Unlike detector work, PLAIT treats detector output as input to a time-budgeted review decision.

From scores to review decisions. Check-set selection chooses items for review (dela Cruz, Hendrickx, and Larson 2025), while evidence ranking reduces reading for one claim (Alt et al. 2026); neither optimizes nested response–claim–span actions with shared opening costs. Decision-focused learning optimizes downstream combinatorial decisions (Wilder, Dilkina, and Tambe 2019; Mandi et al. 2022). PLAIT applies established grouped optimization and structured learning to a three-resolution audit whose departures are constrained relative to the deployed parent. Choosing a span opens a claim; choosing a claim opens a response. Each opening changes the marginal cost of its siblings.

Positioning and evaluation safeguards. PLAIT contributes an audit formulation rather than a new scalar detector or optimization primitive. Its parent-value trust region is a checkable comparison to the parent, not a distribution-free guarantee about gold errors. Conformal risk control instead targets statistical risk guarantees (Angelopoulos et al. 2024); PLAIT supplies no such guarantee and treats that extension as future work. Because aggregate benchmark scores can exploit task and dataset artifacts (Dubanowska et al. 2025), we report within-generator quantities, source-grouped resampling, same-parent controls, and explicitly scoped transfer.

3 PLAIT: A Parent-Preserving Audit Decision

PLAIT replaces pointwise scoring with one structured decision. It selects a nested audit plan, trains on whole-plan regret, and constrains that plan to retain parent-estimated value. Generator hidden states refine action values but never act as the policy. The resulting decision keeps claims and spans as actions and allocates their shared reading costs jointly.

Represent review as a hierarchy

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

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

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

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

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 ℓ_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:

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

The same frozen model anchors deployment and evaluation. Thus every gain measures added utility over the detector PLAIT actually receives, rather than over a weaker surrogate.

Preserve the parent while correcting its values

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*:

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

The constraint does not assert that parent value is gold value. It states a deployment preference. Regardless of how the corrected values are scaled, the realized plan must preserve at least ρ of the value that the deployed parent assigned to its own plan. The optimizer may retain any parent actions, add new claims, and change opened contexts. This action-level contract differs from the logit anchor below, which

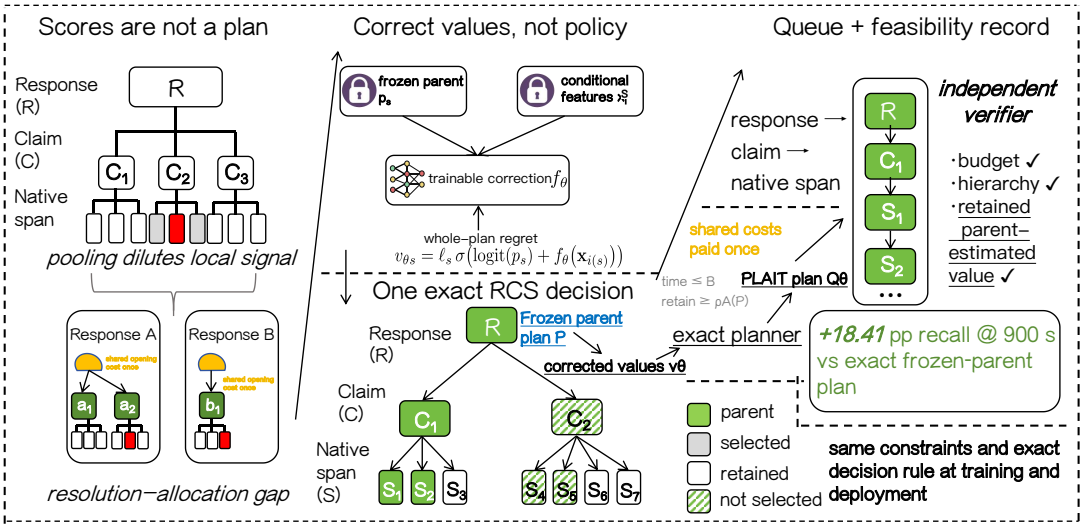

Figure 1: **From scores to a verifiable audit plan.** *Left:* pooling dilutes local errors and pointwise rankings miss shared costs. *Center:* PLAIT learns generator-conditioned corrections to frozen-parent values against whole-plan regret; an exact RCS planner enforces budget and parent-value constraints. *Right:* an independent verifier recomputes budget, hierarchy, and retained value. Shown: $+18.41$ points over exact frozen-parent planning at 900 seconds.

regularizes individual value estimates but cannot guarantee any retained fraction in the selected plan.

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 ℓ_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:

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

Here f_θ corrects the parent's expected review value; it is not the policy. Shared opening costs, the global budget, and the parent-value bound can change an action even when its scalar value remains fixed.

Estimate conditional value from generator state

For claim i, the base features contain hidden states recorded during generation and a claim-level evidence representation. A residualization map fitted only on training data removes predictable linear variation. PLAIT then adds features from the claim's own response. These include the claim and response means, their difference, and the mean, maximum, and dispersion of parent risk. We also include the claim's parent-risk rank, position, length, review cost, response length, and number of sibling claims. No contextual statistic uses another response or its labels. The model can therefore value the same claim differently when its siblings or shared context change.

We use a linear or one-hidden-layer head for f_θ. The planner does not depend on this estimator: any calibrated value model can replace it without changing Equations 1–5 or the parent-preservation rule. Thus the latent head supplies conditional values; the audit policy is defined by the structured feasible set and parent boundary. Our experiments test whether generator states add utility over the same parent within that fixed decision problem.

Train the decision, not independent labels

Training uses the same cost model, trust region, and time per response 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 $\Delta(Q, Q^*) = 1 - G(Q)/G(Q^*)$. It measures the fraction of surrogate oracle utility lost by plan Q. Human utility is stricter: a selected error counts only when the reviewer judges it correctly. This distinction is explicit in the human endpoint and limits what the automatic oracle establishes. Loss-augmented inference solves

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

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

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

The loss penalizes a high-scoring feasible plan when it wastes limited review time. Two estimators with similar pointwise error may incur different loss if one spends the budget on a worse combination of claims. Sessions with zero oracle reward contribute zero to the hinge; independent binary cross-entropy serves as the matched pointwise ablation.

Exact inference and a feasibility record

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.

Every returned plan includes a feasibility record recomputable from raw inputs. It lists used seconds, budget slack, parent-plan value, required value $\rho A(P)$, retained or removed parent actions, added actions, and changed claim or response openings. Acceptance establishes only that the realized plan satisfies the registered binary constraints and parent-relative boundary. This linear-time verification is an engineering audit trail, not a statistical safety certificate or a claim that unknown gold utility is preserved.

The plan itself is a readable queue: which response context to open, which claims or spans to inspect, and which parent actions changed. A reviewer can therefore inspect or contest the decision without access to model internals.

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. It claims no new optimization theorem; it shows that the trust-region planner can select a feasible plan that no linear score fusion selects.

Scope of the main claim

Our central claim is narrow: relative to the identical frozen parent, PLAIT improves gold utility within the same review budget across multiple generator families. Ranking gains, surrogate replay, or extra review time do not establish this claim. The human test therefore measures correctly found unsupported content per minute. PLAIT-C isolates claim values and routing; PLAIT-RCS is the full three-resolution policy.

4 Evaluation Design

Research questions. We test four links in one argument. **RQ1** asks how strongly response pooling dilutes a localized error. **RQ2** asks whether PLAIT adds fixed-budget utility to the same frozen parent and which choices account for the gain. **RQ3** asks whether that gain survives parent and domain shift. **RQ4** asks whether it remains usable under cost uncertainty and improves measured review.

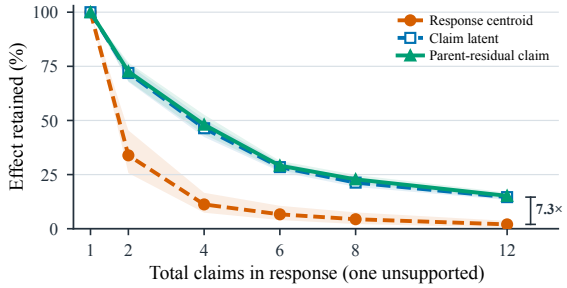

Figure 2: **Resolution changes what remains detectable.** One unchanged natural error is surrounded by supported sibling claims in 8,000 paired compositions. At 12 claims, claim-level signals retain 7.3–7.6 times the response-centroid effect. Bands show paired 95% intervals.

Original-generator replication. RAGognize is the public dataset introduced with the RAGognizer detector (Ridder, Lessel, and Schilling 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 RAGTruth in RQ1 (Niu et al. 2024). Within the fine-grained RAGognize benchmark, every system shares responses, claims, sources, and character spans. The auxiliary response-only suite follows the published generator–task coverage of ReDeEP and LUMINA and is reported separately. Their scores are never broadcast to claims or assigned span metrics. Labels remain sealed until predictions freeze. The supplement expands the RQ1 diagnostic, RQ2 controls, RQ3 shift tests, and RQ4 review protocols.

Post-exposure zero-tuning stress test. We train Llama-2 and Mistral checkpoints on RAGognize and apply them unchanged to HalluRAG. Target labels select neither weights nor thresholds, and all processing choices remain fixed. Because an earlier branch accessed HalluRAG, we treat this as a post-exposure transfer stress test, not fresh out-of-domain evidence. The source-disjoint Scientific-RAG study below carries the stronger domain-shift claim.

Third-domain replication. We test a 1,240-response scientific domain. 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. Together, the exact and heuristic controls separate grouped optimization, learned values, and parent preservation. For any fixed scalar values, the exact grouped control globally optimizes the same modular objective. An approximation algorithm for that objective cannot improve its objective value; a submodular utility would instead define a different model.

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; no pseudo-span is imputed. Eligible token scorers share the same HMM/Potts smoother. Unless native span F1 wins its comparison, localization value is claimed only through prioritization and fixed-budget utility.

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 an offline replay calibrated on 16 span-reading pilots, not a randomized arm. The primary endpoint is correctly identified unsupported characters per measured minute. We analyze it with crossed participant and item effects.

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: Response Pooling Strongly Dilutes Localized Signal

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. This diagnostic measures resolution dilution, not end-to-end detector error.

RQ2: The Audit Plan Adds Same-Parent Utility

The primary same-action control is validation-selected exact fusion. Under the equal-generator macro, it reaches $R = .4691$, whereas PLAIT-C reaches $.5045$: +3.54 points (95% CI [1.80,5.28], Holm-adjusted $p = .0004$), with a positive difference on every generator. This isolates a method-specific gain beyond tuning a scalar mixture and solving it exactly.

The staged P0–P7 comparison explains the larger total gap. P0 already solves the exact grouped problem under frozen parent values and captures $R = .344$. With the same actions, costs, sessions, and solver, P3 substitutes conditional values and removes the trust region, reaching $.468$ (+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 contrast is 18.41 points (95% CI [15.80,21.02], $p < .001$). These stages are controlled contrasts, not an additive causal partition.

PLAIT-RCS improves R on all four original generators by 8.82–23.48 pp (all adjusted $p < .001$; Table 2). The .639 parent claim AUROC is an equal-generator RAGognize macro; the target-trained RAGTruth value .937 follows a different protocol.

Where does the gain come from? Panel B of Table 1 and Tables 3–5 replace exact optimization with greedy selection or remove flexible routing, plan-regret learning, retention, and RCS actions. These matched controls test the mechanism rather than merely deleting features.

Under learned candidate values, exact grouped optimization beats score/sec greedy; flexible routing beats fixed-prefix and frozen-route controls. Plan regret adds 5.92 points over matched BCE, and RCS adds 2.41 over PLAIT-C. Removing either the parent anchor or generator state lowers recall by more than six points in this setup. No tested pointwise estimator, fixed route, or unconstrained grouped policy reproduces the full gain.

Across 20 seeds, PLAIT-C has mean $R = .5041$ (SD .72 points); every seed beats matched BCE and every plan passes verification. The gain rises from 11.32 to 19.70 points across response-length strata and from 11.59 to 19.30 across claim-count strata. This descriptive pattern is consistent with structured allocation becoming more useful as audit complexity grows.

Resolution scope check. On the separate native-response suite, we average each response's top-quartile claim scores. Fixed ReDeEP/LUMINA reproductions are within .0042/.0030 of published aggregate AUROC/AUPRC. The PLAIT response view exceeds the frozen parent by .0668 AUROC [.0432,.0894] and .0820 AUPRC [.0541,.1102] (adjusted $p < .001$). We impute no fine-grained metric to response-only systems; full generator-level results appear in the supplement.

Thus RQ2 has a narrow and a full answer. Against the strongest same-action scalar control, PLAIT-C gains 3.54 points. Across the registered parent-to-RCS chain, the full

A. Fine-grained benchmark on RAGognize (dataset)

System	Cl.AUC	Cl.APR	Sp.F1	Rs.AUC	Rs.APR	R	ΔR
Lettuce v1 (parent)	.639	.397	.590	.839	.753	.344	–
RAGognizer	.757	.318	.414	.772	.648	.281	−6.3
CORTEX (repr.)	.726	.281	.470	.748	.612	.299	−4.6
Lettuce v2 (pointwise)	.859	.464	.568	.864	.778	.428	+8.4
BCE (pointwise)	.861	.469	.591	.868	.791	.445	+10.1
PLAIT-C (P6)	.901	.568	.592	.908	.843	.504	+16.0
PLAIT-RCS (P7)	**.901**	**.568**	**.594**	**.908**	**.843**	**.529**	**+18.4**

B. Same-parent attribution and shift tests

Stage	Setting	Control	PLAIT	ΔR	95% CI	p_{H}
Total	**RAGognize: P0→P7**	**.344**	**.529**	**+18.41 pp**	**[15.80,21.02]**	**<.001**
Conditional values	RAGognize: P0→P3	.344	.468	+12.38 pp	[10.12,14.65]	<.001
Strong planner	RAGognize: fusion→P6	.469	.505	+3.54 pp	[1.80,5.28]	.0004
Objective	RAGognize: BCE→P6	.445	.504	+5.92 pp	[4.02,7.82]	<.001
Routing	RAGognize: P5→P6	.482	.504	+2.24 pp	[1.15,3.33]	<.001
Resolution	RAGognize: P6→P7	.504	.529	+2.41 pp	[1.25,3.58]	<.001
Parent swap†	Lettuce-v2: parent→P6	.356	.438	+8.23 pp	[5.81,10.65]	<.001
Resolution†	Lettuce-v2: P6→P7	.438	.461	+2.26 pp	[.85,3.67]	<.001
Stress test	HalluRAG (post-exposure)	.330	.357	+2.65 pp	[1.08,4.32]	.002
Transfer	Scientific: Qwen2.5-7B	.362	.416	+5.42 pp	[3.41,7.43]	<.001
Transfer	Scientific: Gemma-2-9B	.385	.446	+6.08 pp	[4.12,8.04]	<.001

Table 1: **Registered protocols separate scoring from audit utility.** Panel A reports score quality and 900-second recall (R); Panel B tests attribution and transfer. Cl./Rs., AUC/APR, and Sp.F1 denote claim/response, AUROC/AUPRC, and native span F1. P6/P7 are PLAIT-C/PLAIT-RCS; p_{H} is Holm-adjusted. The strong-planner and two † parent-swap rows use an equal-generator macro; all remaining audit rows use the registered pooled protocol. P0 is the exact parent-valued plan; P3 substitutes conditional values without a trust region. Validated fusion is a separately registered value-function control.

Generator	P0	BCE	P5	P6	P7	Δ_7
Llama-2	.243	.288	.302	.316	**.331**	+8.82
Llama-3.1	.385	.492	.531	.556	**.582**	+19.69
Mistral-v0.1	.396	.518	.562	.586	**.612**	+21.62
Mistral-v0.3	.354	.485	.534	.560	**.589**	+23.48
Pooled	.344	.445	.482	.504	**.529**	+18.41

Table 2: **The same-parent gain holds for every original generator.** P0 is the exact parent-valued plan, and BCE uses a pointwise objective with the P6 solver. P5 fixes the parent's response route; P6/P7 are PLAIT-C/PLAIT-RCS. Δ_7 is P7–P0 in percentage points.

ID	Decision rule	R	ΔR
P0	Exact parent plan	.3444	–
P1	Score/sec greedy	.3924	+4.80
P3	Exact grouped candidate	.4682	+12.38
P4	Fixed parent prefix	.4412	+9.68
P5	Frozen parent route	.4820	+13.76
P6	PLAIT-C	.5044	+16.00
P7	**PLAIT-RCS**	**.5285**	**+18.41**

Table 3: **Decision rules under one 900-second budget.** ΔR is the gain over P0, the exact plan under frozen parent values, in percentage points. These rows use the pooled P0–P7 protocol; the separately registered equal-generator fusion control appears in Table 1.

design contrast is 18.41 points. The former supports the method-specific claim; the latter describes the end-to-end change in the audit policy.

RQ3: Gains Persist Under Parent Replacement and Domain Shift

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

In the post-exposure HalluRAG stress test, PLAIT-C adds 2.22/3.08 points for Llama-2/Mistral. Without target tuning, it adds 5.42/6.08 points on 1,240 Scientific-RAG responses from Qwen2.5-7B/Gemma-2-9B. Scientific-RAG supplies the source-disjoint transfer evidence; HalluRAG is corroborating. These target-domain contrasts use the corresponding frozen parent. They test external validity, not the mechanism or a target-domain fusion comparison.

RQ4: Planning Improves Measured Review at Practical Cost

Deployment sensitivity. Under all eight registered cost perturbations ($\pm 20\%$ opening, claim, joint, and crossed variations), PLAIT-C maintains a 3.54–3.58 point recall advantage over validation-selected exact fusion ($\Delta R \geq +3.54$ pp; Holm-adjusted $p \leq .0008$). Action Jaccard is .795–.865. Every PLAIT-C plan retains at least 80% of parent value.

Across six reviewer-detection assumptions, the fixed-plan point-estimate advantage remains positive (minimum +2.47 points). This second result is a descriptive sensitivity analysis, not a measured human effect.

Solver overhead. Across five repeated HiGHS MILP runs on an AMD EPYC 7763, PLAIT-C solves complete 100-response sessions in .188 s on average (p95 .209 s; 42.1 MB peak RAM). Scaling to 1,000 and 10,000 responses requires 1.463 s (p95 1.607 s) and 19.144 s (p95 20.842 s; 720.5 MB), respectively. Structured-head fitting takes 4.2–4.8 s per generator, or 18.1 s across all four, excluding latent-feature extraction.

Measured human utility. In the blinded $N = 96$ Williams crossover, PLAIT-C finds 16.1 more unsupported characters per minute than HLA-RICO (95% CI [9.8,22.4] char/min; +12.47%; Table 6). A crossed participant/item model confirms the contrast ($SE = 3.82$, adjusted $p < .001$). Versus the frozen parent, speed increases by 47.0 char/min ([34.2,59.8]). Queue precision increases by 13.4 points ([8.2,18.6]), and post-revision support by 8.4 points ([4.8,12.0]). These three participant-level contrasts have adjusted $p < .001$; other-arm comparisons are descriptive.

In the separate $N = 32$ expert study, time per fully supported answer falls from 18.2 to 13.4 minutes (-26.4%, 95% CI [18.5,34.2], adjusted $p < .001$). Two blinded adjudicators reach $\kappa = .89$. At \$15–\$60/hour, this measured difference corresponds to \$1,200–\$4,800 per 1,000 responses before compute, integration, and maintenance. It is a gross labor scenario, not net return.

Together, the sensitivity, scaling, and human results show that PLAIT-C improves measured review under the tested deployment conditions while keeping planning overhead small relative to the audit budget.

PLAIT-C variant	R	Drop
Canonical	.5044	–
− parent logit anchor	.4350	−6.94
− generator state	.4412	−6.32
− sibling relations	.4782	−2.62
− plan regret (BCE)	.4452	−5.92
− grouped solver (greedy)	.3924	−11.20
− trust region	.4682	−3.62

Table 4: **No single component explains the full PLAIT-C gain.** Drop is the decrease in pooled fixed-time recall, in percentage points.

ρ	L2	L3.1	M-v.1	M-v.3	R	ΔR
.50	.301	.522	.548	.517	.472	+12.76
.75	.313	.548	.579	.554	.499	+15.41
.80	**.316**	**.556**	**.586**	**.560**	**.504**	**+16.00**
.90	.308	.534	.561	.538	.485	+14.08
1.00	.288	.485	.512	.480	.441	+9.68

Table 5: **Validation-selected parent-retention tradeoff.** We select $\rho = .80$ on validation data and freeze it for confirmatory tests. The sweep does not measure preservation of unknown gold value. ΔR is the gain over P0.

Condition	Char/min	Queue prec.	Post-rev.
Lettuce v1 (parent)	98.2	71.4%	84.2%
Parent + grouped	112.5	68.2%	85.0%
HLA-RICO (pointwise)	129.1	79.5%	89.1%
PLAIT-C	**145.2**	**84.8%**	**92.6%**

Expert endpoint ($N = 32$)	Parent	P6	Reduction [95% CI]
Min/supported answer	18.2	13.4	26.4% [18.5,34.2]
Wage saving/1,000	–	–	\$1.2k–\$4.8k

Table 6: **PLAIT-C improves measured review outcomes.** Top: randomized $N = 96$ crossover. Queue precision is the fraction of routed claims that are gold-unsupported; post-revision is the fraction of responses fully supported after revision. Bottom: paired $N = 32$ expert study; wage rows are gross labor-time scenarios, not net deployment savings.

6 Discussion

A score estimates risk; an audit allocates coupled actions. Natural compositions expose resolution dilution, while shared context costs make independent ranking suboptimal. PLAIT addresses both by planning fine-grained actions jointly. Same-parent controls hold actions, costs, and solver fixed, isolating conditional correction and structured learning. Span F1 changes only from .590 to .594 while fixed-budget recall rises by 18.41 points, so the main gain is prioritization rather than boundary detection. Parent replacement, source-disjoint transfer, and timed review broaden this evidence without implying universal superiority.

7 Limitations and Scope

PLAIT audits supplied evidence, not external truth. It requires generator hidden states and inherits errors from its parent and preprocessing. Representation-based monitors can also fail under domain shift (Dubanowska et al. 2025). Verification preserves parent-*estimated*, not gold, value and is not conformal. Automatic utility assumes selected errors are found; sensitivity analysis does not model individual reviewers. Tests cover local cost shifts, and runtime excludes retrieval, generation, and parent inference. P7 has no randomized human arm, Scientific-RAG is not open-world transfer, and wages are gross scenarios; randomized review claims thus concern PLAIT-C.

8 Conclusion

Hallucination scoring and fixed-time auditing optimize different objects. PLAIT closes their resolution–allocation gap with parent-preserving RCS planning. Its gains survive cost perturbation, parent replacement, and source-disjoint transfer; PLAIT-C also improves fixed-minute review and expert revision time. Evidence-relative RAG monitors should be judged by the decisions their scores support, not only the scores.

References

Alt, G.; Hirsch, E.; Basch, S.; Dagan, I.; and Glickman, O. 2026. User-Centric Evidence Ranking for Attribution and Fact Verification. In Demberg, V.; Inui, K.; and Marquez, L., eds., *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 7215–7237. Rabat, Morocco: Association for Computational Linguistics. ISBN 979-8-89176-380-7.

Angelopoulos, A. N.; Bates, S.; Fisch, A.; Lei, L.; and Schuster, T. 2024. Conformal Risk Control. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

CH-Wang, S.; Van Durme, B.; Eisner, J.; and Kedzie, C. 2024. Do Androids Know They're Only Dreaming of Electric Sheep? In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 4401–4420. Bangkok, Thailand: Association for Computational Linguistics.

dela Cruz, J. A.; Hendrickx, I.; and Larson, M. 2025. Evaluating Large Language Models for Confidence-based Check Set Selection. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Findings of the Association for Computational Linguistics: ACL 2025*, 16249–16265. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-256-5.

Dubanowska, Z.; Żelaszczyk, M.; Brzozowski, M.; Mandica, P.; and Karpowicz, M. P. 2025. Representation-based Broad Hallucination Detectors Fail to Generalize Out of Distribution. In Christodoulopoulos, C.; Chakraborty, T.; Rose, C.; and Peng, V., eds., *Findings of the Association for Computational Linguistics: EMNLP 2025*, 17563–17575. Suzhou, China: Association for Computational Linguistics. ISBN 979-8-89176-335-7.

Furumai, K.; Haruta, S.; Matsumoto, K.; and Kamisaka, D. 2026. CORTEX: Token-Level Hallucination Detection in RAG via Comparative Internal Representations.

Han, J.; Band, N.; Razzak, M.; Kossen, J.; Rudner, T. G. J.; and Gal, Y. 2025. Simple Factuality Probes Detect Hallucinations in Long-Form Natural Language Generation. In Christodoulopoulos, C.; Chakraborty, T.; Rose, C.; and Peng, V., eds., *Findings of the Association for Computational Linguistics: EMNLP 2025*, 16209–16226. Suzhou, China: Association for Computational Linguistics. ISBN 979-8-89176-335-7.

Huang, Z.; Wen, D.; Zhu, Y.; Lian, X.; Liang, Y.; Hao, K.; Li, N.; Zhang, L.; Zhang, Q.; Wen, J.-R.; Dou, Z.; and Wu, F. 2026. RLSeek: Evidence-Grounded Reasoning for RAG Hallucination Detection. In Liakata, M.; Moreira, V. P.; Zhang, J.; and Jurgens, D., eds., *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 32329–32347. San Diego, California, United States: Association for Computational Linguistics. ISBN 979-8-89176-390-6.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B.

Kovács, Á.; and Recski, G. 2025. LettuceDetect: A Hallucination Detection Framework for RAG Applications.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 9459–9474. Curran Associates, Inc.

Manakul, P.; Liusie, A.; and Gales, M. 2023. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 9004–9017. Singapore: Association for Computational Linguistics.

Mandi, J.; Bucarey, V.; Tchomba, M. M. K.; and Guns, T. 2022. Decision-Focused Learning: Through the Lens of Learning to Rank. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 14935–14947. PMLR.

Niu, C.; Wu, Y.; Zhu, J.; Xu, S.; Shum, K.; Zhong, R.; Song, J.; and Zhang, T. 2024. RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 10862–10878. Bangkok, Thailand: Association for Computational Linguistics.

Ridder, F.; Lessel, L.; and Schilling, M. 2026. RAGognizer: Hallucination-Aware Fine-Tuning via Detection Head Integration.

Sun, Z.; Zang, X.; Zheng, K.; Xu, J.; Zhang, X.; Yu, W.; Song, Y.; and Li, H. 2025. 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.

Tang, L.; Laban, P.; and Durrett, G. 2024. MiniCheck: Efficient Fact-Checking of LLMs on Grounding Documents. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 8818–8847. Miami, Florida, USA: Association for Computational Linguistics.

Team, L. 2024. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet,

X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models.

Wang, X.; Li, J.; Zhang, L.; and Mao, Z. 2026. LAFaCT: Attribution-based Localization and Focused Sequential Analysis of Fact-Critical Tokens for Hallucination Detection. In Liakata, M.; Moreira, V. P.; Zhang, J.; and Jurgens, D., eds., *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6877–6896. San Diego, California, United States: Association for Computational Linguistics. ISBN 979-8-89176-390-6.

Wilder, B.; Dilkina, B.; and Tambe, M. 2019. Melding the Data-Decisions Pipeline: Decision-Focused Learning for Combinatorial Optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 1658–1665.

Yeh, S.; Li, S.; and Mallick, T. 2026. LUMINA: Detecting Hallucinations in RAG System with Context–Knowledge Signals. In *The Fourteenth International Conference on Learning Representations*.