BEYOND RELEVANCE: STATE-CONDITIONED CONSEQUENCE MODELING FOR ADAPTIVE RETRIEVAL-AUGMENTED GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The value of retrieved evidence is not fixed: the same candidate can have different consequences depending on what has already been retrieved. Yet sequential retrieval-augmented generation (RAG) commonly evaluates the next candidate without explicitly modeling this dependence on the accumulated retrieval state. In a canonical singleton-generation audit on HotpotQA and 2WikiMultiHopQA, 33.9% of repeated query–candidate groups change consequence category across retrieval states, confirming that this phenomenon persists after controlling for batch-composition effects.

Motivated by this observation, we formulate next-evidence selection as state-conditioned consequence prediction and introduce the State-Consequence Model (SCM). SCM learns from offline counterfactual supervision produced by a frozen generator to predict the benefit and harm of adding each remaining candidate, while an independent answer verifier determines when retrieval should stop.

A strictly matched state-agnostic ablation degrades both consequence prediction and held-out answer quality without reducing retrieval cost, directly showing that the accumulated state provides useful information for retrieval decisions. Across adaptive RAG evaluations, SCM achieves strong answer quality with efficient retrieval and remains competitive under stricter relevance-matched controls.

Together, these results establish retrieval state as a central variable in predicting the consequence of the next evidence addition and motivate consequence-aware evidence selection for sequential RAG.

1 INTRODUCTION

Retrieval-augmented generation (RAG) combines parametric language models with retrieved evidence and is now a standard design for knowledge-intensive generation and question answering (Guu et al., 2020; Lewis et al., 2020; Karpukhin et al., 2020). Most retrieval pipelines, however, still select a fixed number of passages or preserve a ranking produced before generation. That design is appropriate when retrieval rank is a reliable proxy for downstream usefulness. It becomes brittle in multi-hop settings where candidate pools contain partial evidence, redundancy, lexical distractors, and facts that are individually relevant but interact differently with what the generator has already seen.

The central difficulty is that relevance and downstream consequence are different quantities. A passage may be highly relevant to the question yet have sharply different effects at different retrieval states: it can complete a missing reasoning step, contribute no new information, or introduce a competing fact that overturns an already-correct answer. Prior work establishes that language models are vulnerable to irrelevant context, evidence placement, and retrieval timing (Shi et al., 2023; Liu et al., 2024; Yoran et al., 2024), while newer work explicitly studies generator-aware passage gain or utility (Jiang et al., 2025; Zhang et al., 2025; Deng et al., 2025). Our focus is narrower and sequential: the target is the marginal consequence of the *next insertion* under the current partial state, re-estimated after every state transition.

This distinction is especially important for sequential multi-hop RAG. After one document is added, the next decision is made against a new evidence configuration and often a new current answer. A candidate that was useful early may become redundant or harmful later, while a previously weak

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.

candidate may become the missing bridge. The decision unit is therefore an *insertion action under the current state*, not a document in isolation. Let q be a question, S_k the ordered evidence state after k selected documents, and c a remaining candidate. Conventional reranking estimates a static score $R(q, c)$, or at most a relevance score conditioned on S_k. Adaptive retrieval requires a different quantity: the change in answer value caused by appending c to the current state,

$$R(q, c) \neq \Delta V(q, S_k, c). \tag{1}$$

The distinction is operational rather than semantic. As S_k evolves, q and c can remain fixed while the consequence of insertion changes magnitude or even sign. In a canonical singleton-generation audit, 33.9% of repeated same-(q, c) groups change consequence category across retrieval states, providing a conservative empirical estimate after controlling for batch-composition effects.

This creates two coupled but distinct control questions: *which* remaining document is most likely to improve the current trajectory, and *whether* the current answer is already good enough that further retrieval should stop. Treating both questions as relevance estimation conflates candidate utility with answer preservation. Directly evaluating every candidate by generator intervention would answer the right question but is impractical online: each hypothetical insertion requires another generator call and reference-aware scoring is unavailable. We therefore move interventions offline. SCM learns candidate benefit and harm from $[q; S_k; c]$, while a separate verifier estimates current-answer correctness from $[q; S_k; a_k]$. The deployed controller selects the highest-benefit candidate and combines that signal with verifier confidence in a calibrated stop/add rule; candidate-wise counterfactual generation is never used at inference.

We evaluate the proposal in three stages. First, a question-disjoint canonical audit asks whether the same query–candidate pair changes consequence as the accumulated state changes. Second, a strictly matched state-conditioning ablation compares $[q; c]$ against $[q; S_k; c]$ at both candidate-prediction and frozen held-out system levels. Third, frozen end-to-end experiments compare consequence modeling with practical and strict objective-matched relevance controls and adaptive-RAG baselines. The resulting picture is deliberately bounded: state dependence survives a stricter singleton relabeling protocol, explicit access to S_k significantly improves held-out quality at unchanged retrieval depth, and strict relevance matching shows that the clearest isolated effect of consequence semantics beyond objective choice is retrieval efficiency rather than a guaranteed accuracy gain.

Our contributions are three-fold: (1) we formulate adaptive retrieval around the *state-conditioned marginal consequence of the next candidate insertion*, separating it from static relevance; (2) we introduce SCM, which learns benefit and harm from offline counterfactual supervision and combines candidate scoring with independent current-answer verification; and (3) we provide canonical same-(q, c) state-dependence and stability audits, a strictly matched w/o-state ablation, practical and strict relevance controls, adaptive-RAG comparisons, and component analyses that identify both the gains and the limits of consequence modeling.

2 Related Work

RAG and reranking. RAG grounds generation in retrieved text (Guu et al., 2020; Lewis et al., 2020; Karpukhin et al., 2020). Multi-passage architectures such as Fusion-in-Decoder and Atlas improve evidence aggregation (Izacard & Grave, 2021; Izacard et al., 2023), while cross-encoders improve retrieval quality by jointly encoding a query and candidate passage (Nogueira & Cho, 2019). These methods primarily optimize relevance, likelihood, or static usefulness before the downstream generation state is known. A state-aware reranker can condition on accumulated evidence, but relevance supervision still asks whether a passage matches the information need. SCM instead supervises the change in the frozen generator's answer after the passage is actually inserted, making the target explicitly generator- and state-dependent.

Adaptive retrieval. Iterative RAG methods vary retrieval frequency or depth according to reasoning progress, question complexity, uncertainty, or self-reflection. IRCoT interleaves retrieval with chain-of-thought reasoning (Trivedi et al., 2023); FLARE retrieves from generation-time uncertainty (Jiang et al., 2023); Self-RAG trains retrieval and critique tokens (Asai et al., 2024); Adaptive-RAG predicts question complexity (Jeong et al., 2024); and DRAGIN uses real-time information needs (Su et al., 2024). SeaKR and Stop-RAG introduce self-awareness or explicit stopping control (Yao et al., 2024;

Park et al., 2025). These systems establish that a fixed retrieval budget is often suboptimal. Our question is complementary: under a current accumulated state and a fixed remaining pool, what is the consequence of inserting each candidate, and how should that candidate signal interact with preservation of the current answer?

Context robustness and generator-aware selection. Retrieval can hurt even when a retriever is strong. Irrelevant or conflicting passages can distract the generator (Shi et al., 2023; Yoran et al., 2024), and long-context models can underuse or misweight evidence depending on position (Liu et al., 2024). Recent work already moves beyond topical relevance: GainRAG defines passage "gain" by contribution to correct model outputs (Jiang et al., 2025); utility-based selectors distill LLM utility judgments into smaller passage-selection models (Zhang et al., 2025); and Influence Guided Context Selection values a context through downstream degradation under leave-one-context-out removal from a context list (Deng et al., 2025). We therefore do *not* claim that generator-aware utility is new. The narrower object studied here is sequential: $\Delta V(q, S_k, c)$ is the marginal consequence of appending the next candidate to the current partial state, and it is re-estimated after every state transition. The same-(q, c) reversal audit directly tests this additional state dependence.

3 State-Consequence Modeling

3.1 Problem Formulation

For each question q, a frozen retriever ranks a closed pool $D_q = (d_1, \ldots, d_M)$. The policy maintains an ordered selected state $S_k = (s_1, \ldots, s_k)$ and remaining candidates $C_k = D_q \setminus S_k$, initialized with the top-ranked document. The frozen generator G produces the current short answer

$$a_k = G(q, S_k). \tag{2}$$

For a remaining candidate $c \in C_k$, let $S_k^c = S_k \oplus c$ denote the ordered state after appending c, and let $a_k^c = G(q, S_k^c)$. Given reference answer y, we define an offline answer-value proxy

$$V(a, y) = \mathrm{F1}(a, y) + 0.5\,\mathrm{EM}(a, y), \tag{3}$$

and candidate consequence

$$\Delta V(q, S_k, c) = V(a_k^c, y) - V(a_k, y). \tag{4}$$

Positive ΔV indicates improvement and negative ΔV degradation. Final benchmark performance is still reported with standard EM and token-level F1; V is only an offline supervision/calibration proxy. Consequence is generator-specific rather than an intrinsic document property: another generator, prompt, decoding rule, or accumulated state may respond differently to the same candidate.

For interpretable diagnostics, we map continuous consequence to harmful, stable, and helpful categories using $\epsilon = 0.2$: harmful if $\Delta V \leq -\epsilon$, stable if $-\epsilon < \Delta V < \epsilon$, and helpful if $\Delta V \geq \epsilon$. Exact-match correctness before and after insertion additionally yields $C \to C$, $C \to W$, $W \to C$, or $W \to W$ transitions. Crucially, *state-aware relevance is not consequence-aware*: a relevance model may also consume $[q; S_k; c]$, but its target still rewards semantic or retrieval relevance and cannot directly express two states in which the same candidate is equally relevant yet produces opposite generator transitions.

3.2 Counterfactual Consequence Supervision

The true ΔV is unavailable at deployment because the reference answer is unknown. We therefore construct labels offline with a frozen Qwen2.5-7B-Instruct generator (Qwen Team, 2024). For every observed state–candidate pair, the generator answers twice: once from the current state and once after appending the candidate. Comparing both outputs to the reference produces continuous ΔV, the harm/help category, and the answer-state transition. This supervision directly captures the behavior that an online controller would otherwise need expensive candidate-wise generation to observe.

The formal corpus combines retrieved shallow states with deep rollout states. Shallow examples cover the original retriever distribution, whereas deep rollouts expose states reached only after earlier additions have changed the context and current answer. Deep-priority reconciliation identifies 199,866 exact state–candidate key overlaps between the two sources and yields a mixed corpus of 820,134 rows over 20,000 questions. Mixed supervision is designed for cross-depth coverage rather than to maximize every scalar metric; the ablations in Section 5.4 test this distinction directly.

3.3 State-Consequence Model

SCM is a reusable candidate-consequence predictor; candidate selection and stopping are downstream control operations. Figure 1 separates the shared candidate branch from an independent state-level answer verifier. The current answer is excluded from candidate scoring so that answer correctness is not duplicated once per remaining passage.

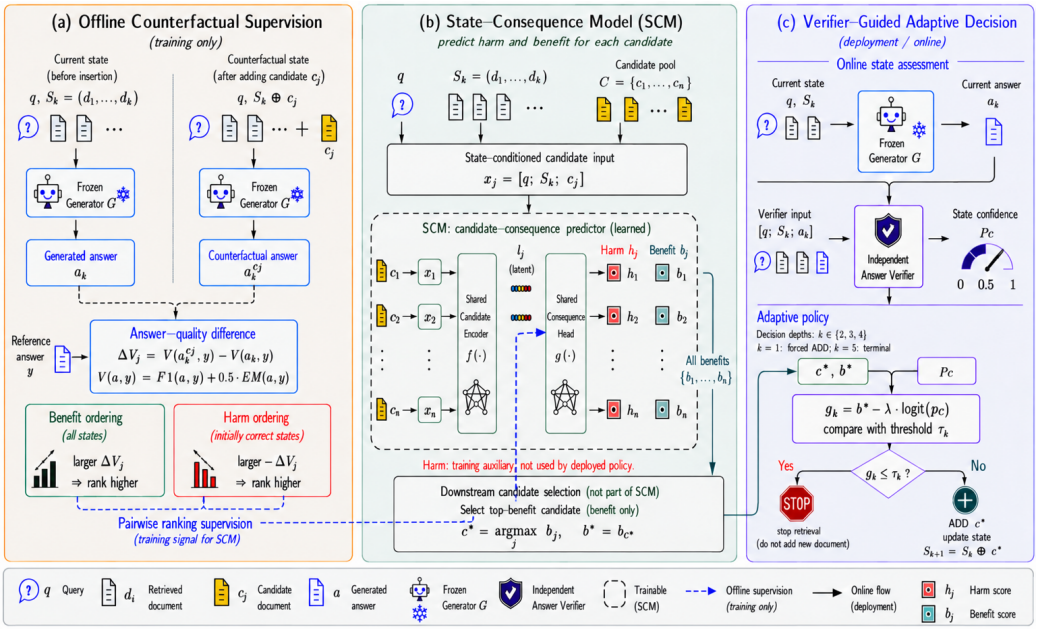

Figure 1: SCM architecture. The candidate branch predicts harm and benefit from $[q; S_k; c_j]$. The independent verifier estimates current-answer correctness confidence p_C from $[q; S_k; a_k]$. Candidate selection uses the maximum predicted benefit b^*, and the downstream depth-aware policy combines (b^*, p_C) to decide STOP versus ADD c^*. Candidate-wise counterfactual generation is used only for offline supervision; online generation is used only for the current answer.

Candidate consequence encoder. Each $x(c) = [q; S_k; c]$ is processed by a shared cross-encoder initialized from the BGE-M3/XLM-R family (Conneau et al., 2020; Chen et al., 2024). A shared MLP outputs benefit $b(q, S_k, c)$ and harm $h(q, S_k, c)$. Separate heads support asymmetric helpful/harmful ranking under strong class imbalance.

State confidence verifier. The verifier receives $u_k = [q; S_k; a_k]$ and predicts $p_C = P(\text{correct} \mid q, S_k, a_k)$. It answers a state-level question—whether the current answer is worth preserving—while the candidate branch asks what is likely to happen if a particular passage is added. We deliberately keep p_C outside the consequence head: candidate selection asks which remaining passage is best; the downstream gate asks whether any passage should be added at all.

3.4 Training Objective and Adaptive Policy

Within-state ranking. Deployment requires the relative order of candidates available in the same state, not calibrated regression to noisy generator-derived ΔV. We therefore optimize pairwise margins. For candidate pairs (i, j) with preferred benefit ordering,

$$\mathcal{L}_b(i,j) = \max(0, m - b_i + b_j). \tag{5}$$

For an initially correct state, define degradation magnitude $r_i^h = -\Delta V_i$ and analogously optimize

$$\mathcal{L}_h(i,j) = \max(0, m - h_i + h_j), \qquad \mathcal{L}_{\text{cand}} = \mathcal{L}_b + \mathcal{L}_h. \tag{6}$$

Checkpoint selection averages harm AP, benefit AP, harm pair accuracy, and benefit pair accuracy on the question-disjoint mixed development set. Transition macro-F1 is diagnostic only, and held-out EM/F1 is never used to select the scorer.

Candidate selection and stopping. At each state,

$$c^* = \arg\max_{c \in C_k} b(q, S_k, c). \tag{7}$$

Exact ties are resolved by original retrieval rank and then by candidate-pool index. Let b_k^* denote the selected benefit logit at depth k. The deployed stop/add score is

$$g_k = b_k^* - \lambda \, \mathrm{logit}(p_C). \tag{8}$$

For $k \in \{2, 3, 4\}$, the policy stops when $g_k \leq \tau_k$ and otherwise appends c^*. Depth 1 forces the first addition and depth 5 is terminal. A shared λ and depth-specific thresholds are fitted on an independent 1,000-question calibration split. The deployed gate uses selected benefit and verifier confidence; harm is learned as an auxiliary consequence signal but is not a direct stop/add term.

4 Experimental Setup

4.1 Datasets and Evaluation Protocol

We evaluate on HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020) using only dataset-provided context pools; no external corpus is added. This closed-pool setting isolates retrieval control from corpus coverage. Mixed training uses 20,000 questions and 820,134 state–candidate rows; candidate selection uses a disjoint 1,000-question mixed development set, stopping calibration another 1,000 questions, and held-out evaluation 2,000 questions (1,260 2Wiki and 740 Hotpot).

Question identity is the isolation unit throughout: rows from the same question never cross training, model development, stopping calibration, and held-out reporting. The 2,000-question held-out set is an internal controlled evaluation drawn from the original dataset partitions rather than a public blind-test leaderboard, so our claims concern paired comparisons under a frozen protocol rather than state-of-the-art benchmark performance.

4.2 Baselines

We include two non-adaptive anchors. **LLM-only** controls for generator parametric knowledge. **Vanilla RAG** follows the original frozen retrieval order with no reranking or adaptive stopping; k is selected from $\{1, \ldots, 5\}$ on development VALUE and frozen at $k = 5$ before test. These anchors separate the value of retrieval itself from adaptive control.

State-aware relevance is our principal practical control. It uses the same candidate input $[q; S_k; c]$ as SCM but a state-static target $r(q, c) = \mathbb{1}[c \text{ is a dataset gold supporting document}]$, with the current answer excluded from candidate scoring. It covers the same 820,134 training and 41,062 development examples, shares the frozen verifier and controller family, and trains relevance with weighted BCE plus within-state ranking. We also add a **strict objective-matched relevance** control: starting from the same precursor, it uses the same benefit-only pairwise ranking objective, margin, examples, update budget, verifier, policy family, and calibration protocol as benefit-only consequence learning. The only intended difference is the within-state ordering signal: binary gold relevance versus generator-derived consequence. Gate parameters are calibrated separately for each scorer on the same independent 1,000-question split.

For RQ2 we additionally train a **strictly matched state-agnostic consequence model**. It uses the same 820,134 training rows, 41,062-row development set, consequence heads, pairwise objectives, update budget, checkpoint-selection rule, verifier, controller family, and independent stopping calibration as Full SCM. The sole intended change is the candidate input: $[q; c]$ instead of $[q; S_k; c]$. This isolates whether explicit access to the accumulated retrieval state improves consequence prediction and downstream control.

We additionally evaluate closed-pool adaptations of Adaptive-RAG (Jeong et al., 2024), FLARE (Jiang et al., 2023), Stop-RAG (Park et al., 2025), and SeaKR (Yao et al., 2024), sharing generator, candidate pool, maximum depth, and answer evaluator. They are controlled adaptations rather than exact reproductions because the original systems use different corpora and retrieval interfaces.

4.3 METRICS AND IMPLEMENTATION

We report EM, token-level F1, the auxiliary $\mathrm{VALUE} = \mathrm{F1} + 0.5\mathrm{EM}$, and average selected documents. Candidate diagnostics include harm/benefit AP, within-state pair accuracy, and transition macro-F1. Statistical comparisons are paired by question because all methods use the same held-out examples.

The frozen generator is Qwen2.5-7B-Instruct with greedy decoding, 4,096 input tokens, and at most 32 new tokens. Maximum depth is five. Candidate inputs use 512 tokens with a 192-token candidate budget and exclude the current answer. Candidate checkpoints are selected only on development data, stop thresholds only on the independent calibration split, and held-out results only after both choices are frozen. Appendix E gives the full frozen configuration and audit trail.

5 EXPERIMENTS AND RESULTS

5.1 RQ1: DOES CANDIDATE CONSEQUENCE DEPEND ON RETRIEVAL STATE?

We first test the core hypothesis directly, before considering any learned stopping policy. We group examples by question and candidate identity and retain candidates that remain available under more than one observed retrieval state. To separate state dependence from execution sensitivity in the original batched FP16 labeling pipeline, we re-evaluate a fixed 1,000-group sample with a canonical singleton protocol: identical prompt construction, fixed document order, greedy decoding, and one prompt per generation call. Query and candidate identity remain fixed; only the accumulated state changes.

Table 1: Same-(q, c) state-dependence audit. The full-population stored-label row quantifies the phenomenon over all 8,000 repeated groups; the canonical singleton row provides a stricter robustness estimate under controlled execution.

Protocol	Repeated (q, c)	Category changes	Change rate	Direct H↔B
Historical stored dev	8,000	3,096	38.70%	363 (4.54%)
Canonical singleton sample	1,000	339	**33.90%**	22 (2.20%)

Across the full set of 8,000 repeated query–candidate groups, 3,096 (38.70%) exhibit at least one change in consequence category across retrieval states under the stored counterfactual labels. Because these labels were originally generated in FP16 batches, we additionally conduct a canonical repeatability audit on a fixed random sample of 1,000 groups using singleton greedy generation with identical serialization and decoding settings. On these same groups, the state-change rate decreases from 39.3% under the stored labels to 33.9% under canonical regeneration. Direct helpful–harmful reversals similarly decrease from 3.9% to 2.2%. Thus, batched execution amplifies the measured rate to some extent, but substantial state dependence remains after removing this source of execution sensitivity.

At the individual state–candidate level, stored and canonical consequence categories agree on 94.21% of 3,609 audited observations. Most disagreements involve a directional label versus the stable category; only five observations (0.14%) switch directly between helpful and harmful. An EM-only canonical audit, using neither VALUE nor the ϵ threshold, still finds state-dependent correctness-transition changes in 26.4% of groups. The paired group-level change-rate difference is -5.4 points (95% CI $[-7.3, -3.5]$), and the direct-reversal difference is -1.7 points (95% CI $[-2.6, -0.8]$). Appendix A.3 gives the full stability analysis, while Appendix A.4 retains the historical threshold sensitivity as a secondary diagnostic.

Figure 2 provides exact-state-matched frozen-generator witnesses rather than population-frequency estimates. Panel (a) shows missing evidence repairing an incorrect answer; panel (b) shows a gold evidence candidate corrupting an already-correct answer; panel (c) holds both q and c fixed and changes only the accumulated state, producing opposite $W \to C$ and $C \to W$ consequences. Together with Table 1, these cases establish the phenomenon that motivates SCM without requiring the stronger claim that every candidate is state-sensitive.

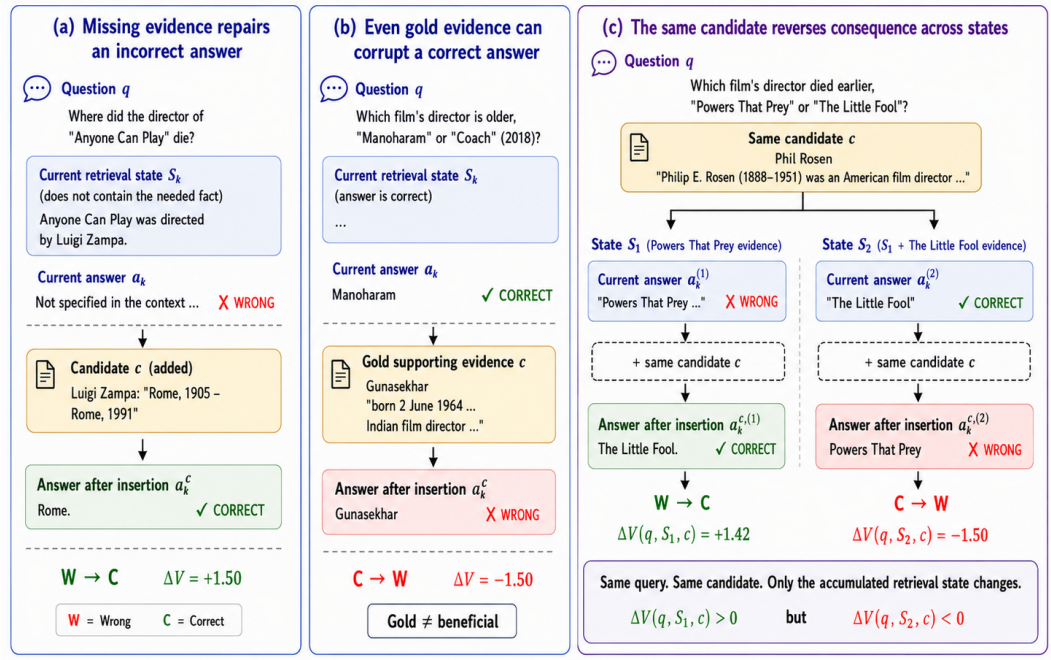

Figure 2: Retrieval consequence is state-dependent rather than determined by relevance alone. (a) Missing evidence repairs an incorrect answer. (b) Even gold evidence can corrupt an already-correct answer. (c) The same query–candidate pair yields opposite counterfactual consequences under two accumulated retrieval states. Examples are drawn from audited frozen-generator trajectories.

5.2 RQ2: Does Explicit State Conditioning Improve Consequence Prediction?

We next isolate the role of the accumulated state itself. Full SCM consumes $[q; S_k; c]$, whereas the matched w/o-state model consumes only $[q; c]$; all training rows, objectives, update budgets, checkpoint selection, verifier, policy family, calibration split, and held-out questions are otherwise held fixed. Table 2 reports both candidate diagnostics and the final frozen policy outcome.

Table 2: Strictly matched state-conditioning ablation. Candidate metrics use the 41,062-row mixed development set; end-to-end metrics use the frozen 2,000-question held-out set.

Metric	Full $[q; S_k; c]$	w/o state $[q; c]$	Full − w/o state
Candidate-level development diagnostics			
Selection score	.4689	.4482	+.0207
Harm AP	.2001	.1641	+.0360
Benefit AP	.3147	.2742	+.0405
Harm pair accuracy	.6417	.6420	−.0003
Benefit pair accuracy	.7191	.7126	+.0065
Transition macro-F1	.2906	.2769	+.0137
Frozen held-out adaptive retrieval			
EM	.7240	.7085	+.0155
F1	.7873	.7713	+.0160
VALUE	1.1493	1.1256	+.0238
Avg. docs	2.3670	2.3665	+.0005

Removing S_k lowers the predeclared candidate selection score from .4689 to .4482, with the largest drops in harm and benefit average precision; static within-group pair ordering partly survives, but state-specific consequence discrimination degrades. The same effect propagates downstream. Full SCM improves held-out EM by 1.55 points (95% CI $[+0.55, +2.55]$, $p = .0038$), F1 by 1.60 points (95% CI $[+0.68, +2.52]$, $p < .001$), and VALUE by .0238 (95% CI $[+.0102, +.0379]$, $p < .001$). Average retrieval depth is unchanged: 2.3670 versus 2.3665 documents (95% CI $[-.020, +.021]$,

$p = 1.00$). This matched ablation therefore connects the RQ1 phenomenon to learned behavior: explicit state conditioning improves consequence-aware decisions rather than simply increasing retrieval budget. Detailed protocol and paired tests appear in Appendix D.1.

5.3 RQ3: Does Consequence Modeling Improve Adaptive RAG?

We next evaluate the complete adaptive system on the frozen 2,000-question held-out set. Table 3 separates three comparison levels: non-adaptive anchors, controlled adaptive-RAG baselines, and the two relevance controls that probe the mechanism most directly.

Table 3: Frozen 2,000-question held-out results; higher EM/F1/VALUE and fewer documents are better.

Method	EM	F1	VALUE	Avg. docs
LLM-only	.3300	.3726	.5376	.0000
Vanilla RAG@5	.6060	.6809	.9839	5.0000
Adaptive-RAG	.6575	.7384	1.0672	2.0395
FLARE	.5430	.6204	.8919	2.9100
Stop-RAG	.6605	.7423	1.0726	2.2035
SeaKR	.6045	.6851	.9874	4.8880
State-aware relevance	.7100	.7803	1.1353	2.5020
Strict matched relevance	.7200	.7839	1.1439	2.3580
SCM (mixed)	**.7240**	**.7873**	**1.1493**	2.3670

Practical and strict controls. Against practical state-aware relevance, SCM improves EM by 1.40 points (95% CI $[+0.45, +2.35]$, $p = .0058$), VALUE by .0140 (95% CI $[+.0010, +.0272]$, $p = .037$), and uses .135 fewer documents ($p < 10^{-4}$); F1 is +.70 points but not significant ($p = .117$). This comparison matches state-conditioned representation and the controller, but not the candidate objective. Under strict objective matching, Full SCM versus strict relevance is +.40 EM, +.34 F1, and +.0054 VALUE, with no significant quality difference ($p = .30$–$.36$) or document-use difference. Benefit-only consequence is likewise quality-tied with strict relevance but uses .088 fewer documents (95% CI $[-.110, -.067]$, $p < 10^{-4}$; Appendix C.2). Thus the clearest isolated effect of consequence supervision is retrieval efficiency, while Full SCM remains numerically strongest in EM/F1.

5.4 RQ4: What Drives Consequence Modeling and Adaptive Stopping?

We separate supervision-source effects on local consequence ranking from deployed-component effects on the final retrieval trajectory (Table 4).

Table 4: Main ablations. † marks a paired difference from Full SCM with $p < 10^{-4}$.

Setting	Selection↑	Harm AP↑	EM↑	F1↑	VALUE↑	Docs↓
Shallow-only	.4510	.1512	72.75	78.84	1.1521	2.519†
Deep-only	**.4749**	**.2042**	72.10	78.21	1.1426	2.387
Mixed / Full SCM	.4689	.2001	72.40	78.73	1.1493	2.367
w/o harm	–	.1604	71.95	78.65	1.1462	2.270†
w/o verifier	.4689	.2001	67.10†	75.37†	1.0892†	2.271†

Supervision source. Deep-only training gives the strongest candidate ranking (.4749), mixed is intermediate (.4689), and shallow-only is weakest (.4510). End-to-end, however, shallow-only and Full SCM are statistically tied in answer quality (VALUE $p = .609$), while Full uses 0.152 fewer documents ($p < 10^{-4}$). Mixed supervision is therefore best interpreted as broader cross-state coverage and a balanced operating point, not as a configuration that must maximize every scalar metric. Local ranking quality and trajectory quality need not move monotonically because a changed ranking also changes which states are subsequently visited.

Deployed components. Removing harm supervision lowers harm AP from .2001 to .1604 but does not significantly change EM/F1/VALUE, so harm is auxiliary under the current benefit-plus-verifier

gate. Removing the verifier is qualitatively different: EM falls by 5.30 and F1 by 3.36 points (both $p < 10^{-4}$). Candidate benefit alone therefore does not protect an already-correct answer; the verifier supplies a distinct preservation signal.

Stopping behavior. Appendix Figure 4 shows that SCM does not succeed by simply retrieving more evidence. Fixed depth 2 is the strongest fixed-depth point, and quality decreases when additional documents are forced. SCM reaches VALUE 1.1493 with 2.367 documents versus 1.0863 at depth 2, while the frozen-trajectory oracle reaches 1.2151 with 2.1995 documents. The remaining oracle gap isolates state-level stopping as a separate bottleneck from candidate ranking.

6 Discussion, Limitations, and Conclusion

The central empirical result is that next-document consequence is not generally static in (q, c). Across all 8,000 repeated groups, the stored-label population statistic is 38.70% (3,096/8,000). A stricter same-sample canonical control shows that the corresponding rate decreases from 39.3% to 33.9% when the fixed 1,000-group sample is regenerated one prompt at a time; direct helpful–harmful reversals decrease from 3.9% to 2.2%. The 94.21% observation-level stored-to-canonical category agreement further shows that execution sensitivity contributes a measurable but limited noise floor rather than explaining the phenomenon away. State conditioning is therefore not universally necessary, but it matters for a substantial subset of sequential decisions. Our novelty is this evolving-state next-action target, not generator-aware utility estimation in general.

The matched w/o-state experiment closes the corresponding model-level question. Removing S_k lowers candidate consequence diagnostics and significantly reduces held-out EM, F1, and VALUE while leaving average retrieval depth unchanged. This isolates a benefit from explicit state conditioning rather than from a larger retrieval budget. Strict relevance matching sharpens a different mechanism claim: practical state-aware relevance is significantly below Full SCM in EM and VALUE, but matching the pairwise objective removes statistically significant answer-quality differences. Consequence semantics therefore should not be credited with the full practical-baseline gap. Its clearest isolated effect beyond objective choice is efficiency: benefit-only consequence retrieves .088 fewer documents than strict relevance at matched answer quality. Harm supervision mainly improves harmful-insertion recognition, whereas the verifier supplies the distinct preservation signal exposed by the large verifier ablation.

The frozen-trajectory oracle reaches 1.2151 VALUE versus 1.1493 realized VALUE, leaving substantial headroom in stopping. Limitations include generator-specific labels, two QA datasets, closed 10-document pools, depth five, one frozen generator, and expensive offline counterfactual construction. The canonical audit also shows that batched FP16 execution can introduce small label sensitivity even under greedy decoding, motivating explicit execution audits in future counterfactual-supervision pipelines. Future work should test cross-generator and open retrieval settings; overall, SCM shifts adaptive retrieval from static relevance to predicted next-insertion consequence.

AI Use Statement

The authors used OpenAI ChatGPT to assist with literature exploration, research brainstorming, feedback on methodology and experimental design, code drafting and debugging, result interpretation, figure and layout refinement, and manuscript drafting and editing. All AI-assisted code, analyses, figures, references, and claims were reviewed and verified by the authors against executed experiments and source materials. The authors take full responsibility for the final content of this work.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. BGE M3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*, 2020.

Jiale Deng, Yanyan Shen, Ziyuan Pei, Youmin Chen, and Linpeng Huang. Influence guided context selection for effective retrieval-augmented generation. *arXiv preprint arXiv:2509.21359*, 2025.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: Retrieval-augmented language model pre-training. In *Proceedings of ICML*, 2020.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of COLING*, 2020.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of EACL*, 2021.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43, 2023.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of NAACL-HLT*, 2024.

Yi Jiang, Sendong Zhao, Jianbo Li, Haochun Wang, and Bing Qin. GainRAG: Preference alignment in retrieval-augmented generation through gain signal synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10746–10757, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.527.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of EMNLP*, 2023.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

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

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*, 2019.

J. Park, S. Cho, and J.-Y. Lee. Stop-RAG: Value-based retrieval control for iterative RAG. *arXiv preprint arXiv:2510.14337*, 2025.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Proceedings of ICML*, 2023.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of ACL*, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of ACL*, 2023.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP*, 2018.

Z. Yao, W. Qi, L. Pan, S. Cao, L. Hu, W. Liu, L. Hou, and J. Li. SeaKR: Self-aware knowledge retrieval for adaptive retrieval augmented generation. *arXiv preprint arXiv:2406.19215*, 2024.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In *International Conference on Learning Representations*, 2024.

Hengran Zhang, Keping Bi, Jiafeng Guo, Jiaming Zhang, Shuaiqiang Wang, Dawei Yin, and Xueqi Cheng. Distilling a small utility-based passage selector to enhance retrieval-augmented generation. *arXiv preprint arXiv:2507.19102*, 2025. Accepted by SIGIR-AP 2025.

A Counterfactual Supervision, Data Integrity, and State-Dependence Audits

A.1 Supervision Sources and Label Construction

SCM is trained from two complementary state sources. Shallow retrieved-state examples cover broad evidence configurations produced from the frozen retrieval order. Deep rollout examples follow sequential additions and therefore expose states that occur after earlier candidate choices have already changed the evidence available to the generator. The final mixed corpus uses deep-priority reconciliation so that exact duplicated state–candidate inputs receive a single target.

Table 5: Supervision-source composition.

Source	Questions	Rows	Role
Retrieved-state source	20,000	420,000	Broad shallow coverage
Deep rollout source	20,000	600,000	Sequential post-addition states
Retained retrieved rows	20,000	220,134	Non-conflicting shallow contribution
Final mixed corpus	20,000	820,134	Formal SCM continuation corpus

The deep source contributes 378,000 rows from 2WikiMultiHopQA and 222,000 from HotpotQA. Its depth distribution is 180,000/160,000/140,000/120,000 rows at state depths 1/2/3/4. For every state S_k and remaining candidate c, the frozen generator is evaluated twice offline: $a_k = G(q, S_k)$ and $a_k^c = G(q, S_k \oplus c)$. Counterfactual generator calls are never required by the deployed policy.

Table 6: Consequence-category distribution in the 600k deep corpus.

Category	Rule ($\epsilon = 0.2$)	Rows	Share
Helpful	$\Delta V \geq +0.2$	41,861	7.0%
Stable	$-0.2 < \Delta V < +0.2$	495,074	82.5%
Harmful	$\Delta V \leq -0.2$	63,065	10.5%

The severe stable-class dominance is why raw three-way accuracy is not used for model selection. Candidate diagnostics emphasize average precision for the minority helpful/harmful classes, within-state pair accuracy, and transition macro-F1.

A.2 Key-Level Reconciliation and Question-Level Isolation

Shallow and deep generation paths can reach the same state–candidate key (q, S_k, c). Because the current answer is excluded from the candidate-scoring input, reconciliation is performed on this key rather than on the generated answer. We emphasize that a key overlap does *not* imply byte-identical generator execution: the shallow and deep sources were produced independently, and batched FP16 execution can differ in batch composition even under greedy decoding. We find 199,866 overlapping state–candidate keys. Among them, 177,385 also have matching current answers, while 22,481 have different current answers across the independently generated sources. Within the 177,385 current-answer-matched overlaps, 170,362 have matching consequence categories, 5,879 have category conflicts, and 1,144 differ only in continuous ΔV. We retain the deep rollout row for every overlapping state–candidate key, yielding the final 820,134-row mixed corpus. The canonical singleton audit below directly quantifies how much this execution sensitivity affects the main state-dependence statistic.

Question identity is the isolation unit throughout the study. Candidate rows from the same question are never split independently across training, checkpoint selection, stopping calibration, and held-out reporting.

The held-out set contains 1,260 2WikiMultiHopQA and 740 HotpotQA questions. It is an internal, frozen, question-disjoint evaluation drawn from the original dataset partitions, not an official public blind-test leaderboard. Calibration uses only its independent 1,000-question split and never the 2,000 held-out questions.

Table 7: Mixed-corpus overlap audit and deep-priority reconciliation.

Audit item	Count
Exact state–candidate key overlaps	199,866
Current-answer matches within overlaps	177,385
Current-answer mismatches within overlaps	22,481
Matching consequence categories (answer-matched)	170,362
Consequence-category conflicts (answer-matched)	5,879
Continuous-ΔV-only differences (answer-matched)	1,144
Final mixed training rows	820,134

Table 8: Question-disjoint experimental partitions.

Partition	Questions	Rows/states	Purpose
Mixed train	20,000	820,134 rows	Optimize candidate scorer
Mixed dev	1,000	41,062 rows	Checkpoint selection
Stop calibration	1,000	5,000 states	Fit depth-aware gate
Held-out evaluation	2,000	Frozen trajectories	End-to-end reporting

A.3 Canonical Singleton Stability Audit

To separate retrieval-state dependence from execution sensitivity, we sample 1,000 repeated same-(q, c) groups from the question-disjoint deep development audit (seed 3401) and regenerate every required state and state-plus-candidate prompt one at a time with Qwen2.5-7B-Instruct, FP16, greedy decoding, the same serialization, fixed document order, 4,096-token input limit, and 32-token output limit. This canonical singleton protocol removes batch-composition variation from the comparison.

Table 9: Historical stored labels versus canonical singleton relabeling on the same 1,000 repeated-(q, c) groups.

Audit quantity	Stored sample	Canonical singleton
Consequence-category change	39.3%	**33.9%**
Direct helpful↔harmful	3.9%	**2.2%**
EM-only transition change	–	26.4%
Direct EM $W \leftrightarrow C$	–	1.3%

At the individual state–candidate level, stored and canonical consequence categories agree on 94.21% of 3,609 audited observations, and EM labels agree on 95.07%. Most disagreements involve transitions between a directional label and the stable category rather than a reversal of consequence direction: only five observations (0.14%) switch directly between helpful and harmful. At the group level, canonicalization reduces the category-change rate by 5.4 points (paired 95% CI $[-7.3, -3.5]$) and the direct helpful–harmful reversal rate by 1.7 points (95% CI $[-2.6, -0.8]$). Thus the original batched labels modestly overstate the magnitude of state dependence, but the primary conclusion remains strong under a stricter execution protocol. Because greedy decoding is deterministic conditional on the realized numerical computation, we attribute the discrepancy to execution sensitivity in independently batched FP16 generation rather than to sampling stochasticity. We therefore retain the original large-scale supervision while using canonical singleton regeneration for the robustness analysis.

A.4 Sensitivity to the Consequence Definition

Table 10 reports sensitivity of the *historical stored-label* audit to the consequence definition. The category-change rate remains above 37% throughout $\epsilon \in [0.10, 0.30]$, and changing the EM weight from 0 to 1 changes the rate by less than one percentage point. An historical EM-only correctness-transition audit also finds changes in 30.76% of repeated query–candidate groups. These rows are retained as a broad threshold diagnostic; the full-population stored-label rate remains the population

statistic, while Table 9 provides the stricter same-sample robustness estimate used to calibrate its magnitude.

Table 10: Sensitivity of the question-disjoint repeated-(q, c) state-dependence audit. Direct reversal denotes helpful↔harmful for thresholded consequence and $W \leftrightarrow C$ for the EM-only audit.

Definition	Setting	Change rate	Direct reversal
$V = \mathrm{F1} + 0.5\,\mathrm{EM}$	$\epsilon = .10$	41.01%	5.39%
	$\epsilon = .15$	39.86%	4.96%
	$\epsilon = .20$	38.70%	4.54%
	$\epsilon = .25$	37.89%	4.23%
	$\epsilon = .30$	37.18%	3.82%
$V = \mathrm{F1} + \alpha\,\mathrm{EM}$	$\alpha = 0$	37.86%	4.12%
	$\alpha = .25$	38.70%	4.54%
	$\alpha = .50$	38.70%	4.54%
	$\alpha = .75$	38.70%	4.54%
	$\alpha = 1.0$	38.70%	4.54%
EM-only transition	no threshold	30.76%	2.51%

A.5 Additional Audited State-Dependence Cases

The main text uses only three cases so that the figure remains readable. Figure 3 adds three strict same-candidate reversals and two harmful insertions from exact-state-aligned frozen-generator logs. These examples document observed generator-specific counterfactual outcomes; they should not be interpreted as revealing an unobserved internal causal mechanism of the generator.

B Model Training, Checkpoint Selection, and Policy Calibration

B.1 Candidate Scorer and Optimization

The final candidate scorer is a continuation of a shallow retrieved-state precursor (called "Phase-4" only in experiment logs). The precursor is initialized from BGE-M3 and trained on the 420,000-row, 20,000-question retrieved-state consequence corpus with the same within-state harm/benefit ranking formulation. Its epoch-2 checkpoint is frozen on a fixed train-derived calibration split by benefit-best trajectory VALUE (1.06799); clean development and held-out questions are excluded. Every deep/mixed continuation and matched-budget source ablation starts from this same checkpoint.

Table 11: Formal mixed-continuation training configuration.

Item	Setting
Initialization	Shallow retrieved-state precursor, epoch 2
Learning rate	5×10^{-6}
Weight decay	0.01
Seed	2028
Groups per batch	2
Gradient accumulation	4
Precision	fp16
Gradient checkpointing	enabled
Pairwise margin	0.1
Harm / benefit rank loss weights	1.0 / 1.0
BCE / continuous-ΔV loss	0 in final continuation

Epoch 1 is selected by the predeclared four-metric selection score. Transition macro-F1 and held-out EM/F1 are not selection criteria. The mixed continuation runs for two epochs at learning rate 5×10^{-6} with weight decay .01, warmup ratio .06, dropout .1, seed 2028, fp16, gradient checkpointing, two state groups per batch, four-step gradient accumulation, and pairwise margin .1. The candidate input cap is 512 tokens with a 192-token candidate budget; the current answer is excluded.

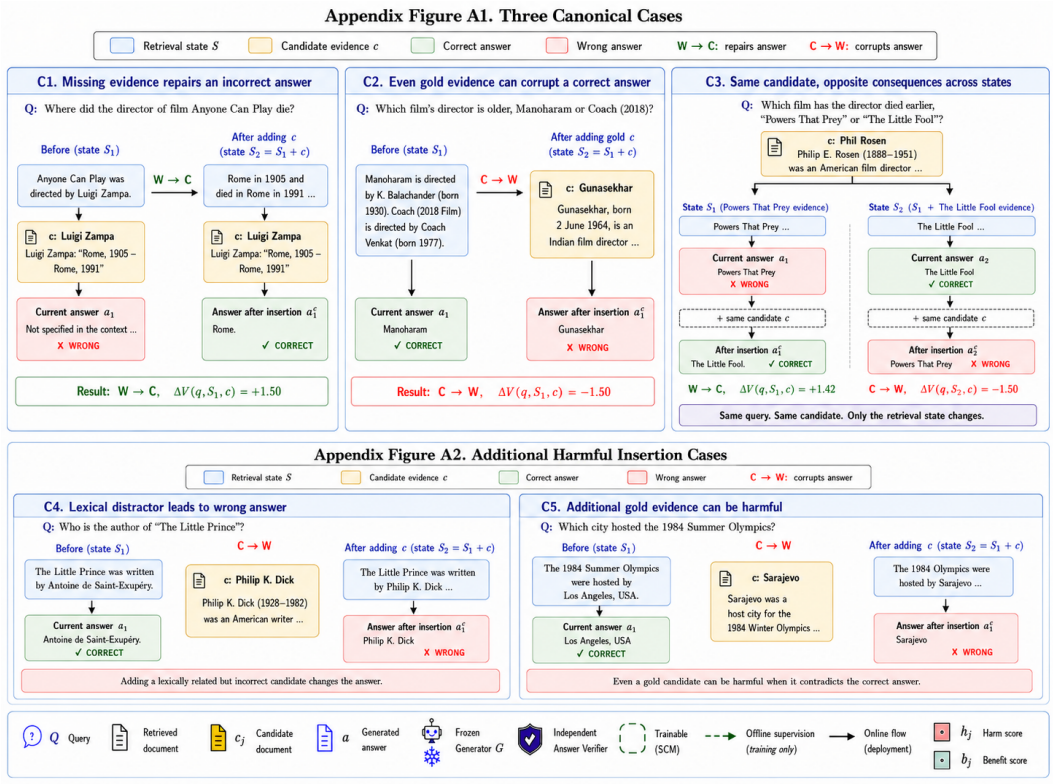

Figure 3: Additional audited consequence cases. Top: strict same-candidate reversals where the query and candidate are fixed while the retrieval state changes. Bottom: harmful insertions, including a lexical distractor and a gold-evidence candidate.

Table 12: Standardized mixed-dev checkpoint comparison.

Checkpoint	Harm AP	Benefit AP	Harm pair	Benefit pair	Selection	Macro-F1
Mixed epoch 1	.200079	.314628	.641559	.719256	.468880	.290493
Mixed epoch 2	.194919	.300750	.654146	.722750	.468141	.241956

B.2 Independent Answer Verifier

The verifier is trained independently from the candidate scorer. State records are deduplicated by (dataset, question id, state name); each record serializes the question, accumulated state documents, and current generated answer, with the binary target given by whether that current answer is exact-match correct before another insertion. Training uses 7,000 questions (top-1/top-3/top-5 state records; 21,000 unique states) and a disjoint 1,000-question development set (3,000 states). The encoder is initialized from BGE-M3 and optimized with binary cross-entropy for five epochs using learning rate 10^{-5}, weight decay .01, warmup ratio .06, dropout .1, batch size 4, gradient accumulation 4, fp16, gradient checkpointing, and seed 2031. Training truncates verifier inputs to 384 tokens and the current answer to 64 tokens; the deployment wrapper permits up to 512 tokens.

The frozen verifier checkpoint is epoch 2. On its 3,000-state development set it obtains 83.2% accuracy, .8417 F1, .9157 ROC-AUC, and .9075 average precision. We do not separately probability-calibrate p_C: the raw verifier probability enters Equation 8, while its relative scale is absorbed by λ and the depth-specific thresholds fitted only on the independent 1,000-question stopping-calibration split.

B.3 Frozen Stopping Policy

The calibrated Full SCM policy uses $\lambda = 4.0$ and depth-specific thresholds shown below. The policy family and lambda grid were fixed before held-out evaluation.

Table 13: Frozen stopping parameters.

Parameter	Value
λ	4.0
τ_2	24.1345962542
τ_3	29.4251170989
τ_4	29.2265299578
Decision depths	2, 3, 4
Forced / terminal	depth 1 adds; depth 5 terminal

C Baseline Controls

C.1 Matched State-Aware Relevance Control and Significance

The matched relevance control uses the same state-conditioned candidate input and a matched candidate encoder, but its supervision semantics and corresponding training objective differ from SCM. For each candidate, the relevance target is static across retrieval states: $r(q, S_k, c) = r(q, c) = \mathbb{1}[c \text{ is a dataset gold supporting document}]$. Gold labels are propagated to deep states by dataset, question identity, and candidate identity. The formal relevance corpus therefore covers the same 820,134 mixed training rows and 41,062 development rows as the SCM comparison, with 47,775 positive training labels and 2,407 positive development labels.

Table 14: Matched state-aware relevance control. The comparison matches state-conditioned representation and the downstream decision framework while deliberately changing supervision semantics and the corresponding candidate objective.

Component	State-aware relevance	SCM
Candidate input	$[q; S_k; c]$	$[q; S_k; c]$
Current answer in candidate scorer	Excluded	Excluded
Candidate encoder	Matched state-conditioned encoder	Matched state-conditioned encoder
Training state–candidate examples	820,134 mixed rows	820,134 mixed rows
Development examples	41,062 rows	41,062 rows
Supervision target	Gold supporting-document membership	Generator-derived insertion consequence
Target changes with S_k for fixed (q, c)?	**No**	**Yes, potentially**
Candidate objective	Weighted pointwise BCE + within-state relevance ranking	Harm/benefit consequence ranking
Harm head	Disabled	Enabled
Answer verifier	Same frozen verifier	Same frozen verifier
Candidate pool / max depth	Same	Same
Stop-calibration questions	Same 1,000-question split	Same 1,000-question split
Gate parameters	Calibrated separately	Calibrated separately
Held-out evaluation	Same 2,000 questions	Same 2,000 questions

Table 15 reports paired question-level tests on the frozen 2,000-question held-out set.

Table 15: Paired Full SCM minus matched state-aware relevance. Confidence intervals use 20,000 paired bootstrap replicates; p-values use 20,000 paired sign-flip replicates.

Metric	Difference	95% CI	p
EM	+1.40 pt	$[+0.45, +2.35]$ pt	.0058
F1	+0.70 pt	$[-0.16, +1.57]$ pt	.1173
VALUE	+.0140	$[+.0010, +.0272]$	.0371
Avg. docs	-.135	$[-.174, -.0955]$	$< 10^{-4}$

The matched comparison therefore supports a statistically detectable gain in EM and VALUE together with lower retrieval cost, while the F1 improvement should be read as positive but inconclusive rather than significant. Because this practical control does not match every output head or loss term, it rules out state conditioning alone but does not isolate supervision semantics from optimization.

C.2 Strict Objective-Matched Relevance Control

To isolate supervision semantics more tightly, we compare *benefit-only consequence* against a strict relevance model with the same precursor checkpoint, $[q; S_k; c]$ input, single deployed benefit score, pairwise ranking loss, margin, 820,134/41,062 train/dev examples, 14,646-update primary budget, verifier, policy family, and independent calibration protocol. The ordering target is the controlled difference: generator-derived ΔV for consequence learning versus binary gold-support relevance for strict relevance. Each scorer receives its own calibration under the same fixed grid because logit scales differ.

Table 16: Strict objective-matched control on the frozen 2,000-question held-out set.

Method	EM	F1	VALUE	Avg. docs
Strict matched relevance	.7200	.7839	1.1439	2.358
Benefit-only consequence	.7195	.7865	1.1462	**2.270**
Full SCM	**.7240**	**.7873**	**1.1493**	2.367

Table 17: Paired tests for the strict control. Differences are row-1 method minus strict matched relevance; CIs use 20,000 paired bootstrap replicates and p-values use 20,000 paired sign-flip replicates.

Comparison	Metric	Difference	95% CI	p
Benefit-only − strict	EM	$-.05$ pt	$[-.80, +.70]$ pt	1.000
	F1	$+.255$ pt	$[-.424, +.935]$ pt	.458
	VALUE	$+.0023$	$[-.0081, +.0127]$	.658
	Avg. docs	$-.088$	$[-.110, -.067]$	$< 10^{-4}$
Full SCM − strict	EM	$+.40$ pt	$[-.35, +1.15]$ pt	.358
	F1	$+.343$ pt	$[-.337, +1.033]$ pt	.328
	VALUE	$+.0054$	$[-.0048, +.0158]$	.300
	Avg. docs	$+.009$	$[-.0125, +.0300]$	.430

Under strict objective matching, benefit-only consequence and strict relevance are statistically indistinguishable in EM, F1, and VALUE, while consequence supervision retrieves .088 fewer documents on average (95% CI $[-.110, -.067]$, $p < 10^{-4}$). Full SCM achieves the highest numerical answer quality, but its gains over strict relevance are not statistically significant. This control shows that part of the larger gap against the practical relevance baseline is attributable to the training formulation; the clearest isolated effect of consequence supervision here is retrieval efficiency.

C.3 Non-adaptive Baseline Anchors

LLM-only uses the frozen Qwen2.5-7B-Instruct generator with no retrieved context. Vanilla RAG uses the original frozen retrieval order with no SCM reranking or adaptive stopping. We select $k \in \{1, 2, 3, 4, 5\}$ on the 1,000-question mixed development split by VALUE. Development VALUE is 0.6056/0.8829/0.9402/0.9493/0.9558 for $k = 1/2/3/4/5$, so $k = 5$ is frozen before held-out evaluation. SCM minus Vanilla RAG@5 yields +11.80 EM points (95% CI $[+10.05, +13.60]$), +10.64 F1 points (95% CI $[+9.06, +12.28]$), +0.1654 VALUE (95% CI $[+0.1410, +0.1902]$), and -2.633 documents (95% CI $[-2.6705, -2.5945]$); all paired randomization tests give $p < 10^{-4}$.

D Additional Ablations and Diagnostics

D.1 Matched State-Conditioning Ablation

The state-conditioning ablation is trained on exactly the same 820,134 mixed training rows as Full SCM and evaluated on the same 41,062-row mixed development set. It uses the same initialization,

consequence heads, pairwise harm/benefit ranking objectives, update schedule, checkpoint-selection score, frozen answer verifier, candidate pool, maximum depth, controller family, and independent 1,000-question stopping-calibration split. The only candidate-side change is removal of the accumulated state from the scorer input: Full SCM uses $[q; S_k; c]$, whereas w/o-state uses $[q; c]$. The stateless model selects epoch 2 with development selection score .44823; Full SCM selects epoch 1 with .46889.

Table 18: Paired Full SCM minus w/o-state on the frozen 2,000-question held-out set. Confidence intervals use 20,000 paired bootstrap replicates and p-values use 20,000 paired sign-flip replicates.

Metric	Difference	95% CI	p
EM	+1.55 pt	$[+0.55, +2.55]$ pt	.0038
F1	+1.60 pt	$[+0.68, +2.52]$ pt	$< .001$
VALUE	+.0238	$[+.0102, +.0379]$	$< .001$
Avg. docs	+.0005	$[-.020, +.021]$	1.000

The absence of any retrieval-depth difference is important: the quality gain cannot be explained by a policy that simply retrieves more evidence. Candidate-level diagnostics also localize the effect. Removing state lowers harm AP from .2001 to .1641 and benefit AP from .3147 to .2742, while harm pair accuracy remains nearly unchanged (.6417 versus .6420). This pattern is consistent with a stateless model retaining some static candidate ordering while losing state-specific consequence discrimination.

D.2 Verifier Ablation and Mechanism

The verifier-free ablation corresponds to the $\lambda = 0$ endpoint in the original predeclared additive-logit calibration sweep; it is not a separately tuned post-hoc policy.

Table 19: Full SCM versus SCM without the answer verifier.

Policy	EM	F1	VALUE	Avg. docs
Full SCM	.7240	.7873	1.1493	2.367
Without verifier	.6710	.7537	1.0892	2.271
Difference	+.0530	+.0336	+.0601	+.096

Paired 20,000-replicate bootstrap confidence intervals and 20,000-replicate sign-flip tests give $p < 10^{-4}$ for EM, F1, VALUE, and average documents. The verifier-free policy can repair some initially wrong depth-2 states but more often fails to preserve answers that are already correct, establishing that p_C is not merely an early-stop heuristic.

D.3 Fixed-Depth and Oracle Controls

Table 20: Fixed-depth controls and calibrated stopping.

Policy	EM	F1	VALUE	Avg. docs
Depth 1	.3235	.3795	.5413	1.000
Depth 2	.6700	.7513	1.0863	2.000
Depth 3	.6585	.7369	1.0662	3.000
Depth 4	.6450	.7283	1.0508	4.000
Depth 5	.6195	.6968	1.0066	5.000
SCM	.7240	.7873	1.1493	2.367
Oracle on frozen SCM trajectory	.7645	.8329	1.2151	2.1995

Depth 2 is the strongest fixed-depth policy. Beyond depth 2, answer quality decreases monotonically even though more documents are provided. The oracle chooses, for each frozen SCM candidate trajectory, the reference-best depth; it is an upper-bound diagnostic rather than a deployable baseline.

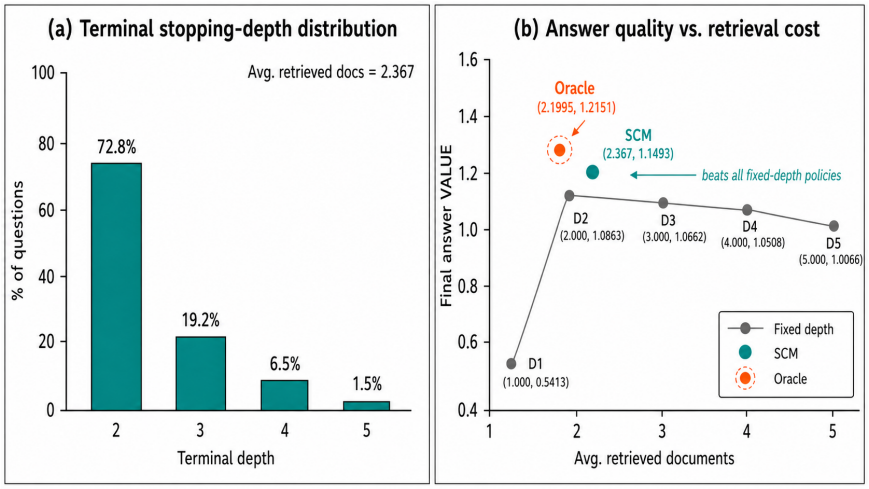

Figure 4: Adaptive stopping behavior. (a) Terminal stopping-depth distribution of Full SCM on the frozen held-out set. (b) Final answer VALUE versus average retrieved documents for fixed-depth policies, Full SCM, and oracle stopping. SCM exceeds every fixed-depth operating point while using 2.367 documents on average; oracle stopping is an upper-bound diagnostic rather than a deployable policy.

D.4 Efficiency Protocol

Latency is measured on one physical NVIDIA RTX A6000 with no concurrent workload. Ten disjoint warm-up questions are excluded. The measured subset contains 100 questions (63 2WikiMultiHopQA and 37 HotpotQA). Each method runs three complete frozen permutations. Timing excludes process/model/dataset loading but includes online query construction, candidate/retrieval scoring, stop decisions, and final generation. CUDA is synchronized immediately before and after each question.

Table 21: Mean online latency on the frozen 100-question subset.

Method	Seconds / question
FLARE	.7639
Adaptive-RAG	1.0914
Stop-RAG	1.2559
SCM	1.5309
SeaKR	45.2383

SCM raw run means are 1.5103, 1.5356, and 1.5445 seconds/question, and all three runs produce identical decision trajectories and call counts. SCM is therefore not presented as the fastest controller; its efficiency claim is narrower: it avoids candidate-wise online counterfactual generation while remaining far cheaper than the controlled SeaKR adaptation.

D.5 Matched-Budget Supervision and Harm-Head Ablations

The supervision-source ablation avoids a confound in which a corpus receives more optimizer updates simply because it contains a different number of groups. Shallow, deep, and mixed continuations share the same shallow-precursor epoch-2 initialization, optimizer settings, and final update budget of 29,292. Only the predeclared 14,646- and 29,292-update milestones are eligible for standardized mixed-dev evaluation.

Shallow and deep continuation are selected only by the same four-metric mixed-dev score used for Full SCM. Both select the 14,646-update milestone; Full mixed likewise selects 14,646 updates. No held-out result is used for checkpoint selection. Deep-only improves the candidate selection score from .4510 (shallow) to .4749, whereas Mixed reaches .4689. Removing harm supervision

Table 22: Matched-budget continuation schedule.

Source	Rows	Groups	Updates / epoch
Shallow	420,000	60,000	7,500
Deep	600,000	80,000	10,000
Mixed	820,134	117,162	14,646

leaves benefit ranking essentially unchanged but lowers harm AP from .2001 to .1604 and harm-pair accuracy from .6416 to .6212. The supported role of harm supervision is therefore improved harmful-insertion discrimination rather than a guaranteed downstream QA gain.

Table 23: Frozen held-out end-to-end results for supervision-source and harm-head ablations.

Variant	EM	F1	VALUE	Avg. docs
Shallow-only	.7275	.7884	1.1521	2.519
Deep-only	.7210	.7821	1.1426	2.387
Full mixed SCM	.7240	.7873	1.1493	2.367
Benefit-only (w/o harm)	.7195	.7865	1.1462	2.270

Shallow-only and Full SCM are statistically indistinguishable in EM, F1, and VALUE, but Full uses significantly fewer documents (2.367 versus 2.519). Benefit-only likewise does not differ significantly from Full in answer quality and uses fewer documents. These results motivate the balanced cross-state interpretation of mixed supervision used in the main text.

D.6 Training-Question Scale Diagnostic on the Shallow Precursor

Before the final deep/mixed continuation, we tested whether simply increasing the number of shallow retrieved-state training questions was sufficient to improve downstream quality. The 20k set is nested: the original 7k training questions are retained and 13k new questions are added, while the clean development protocol is fixed. This diagnostic belongs to the shallow precursor rather than the final mixed SCM.

Table 24: Training-question scale diagnostic for the shallow retrieved-state precursor on the same clean development comparison. Increasing from 7k to 20k questions does not improve answer quality.

Training questions	EM	F1	VALUE	Stop rate
7k	.485829	.600927	.843842	.2206
20k	.482125	.599126	.840188	.0700

The paired VALUE difference is $-.003653$ for 20k minus 7k (95% CI $[-.006493, -.000783]$). HotpotQA is essentially unchanged (VALUE .967981 versus .967903), while the small aggregate decrease is driven mainly by 2WikiMultiHopQA. This negative scaling result cautions against attributing the final system behavior to question count alone and motivated our later emphasis on sequential state coverage rather than simply adding more shallow examples.

E REPRODUCIBILITY CHECKLIST

Table 25: Frozen settings required to reproduce the reported main results.

Item	Frozen setting
Retriever / candidate pool	BGE-M3-based frozen retrieval over dataset-provided 10-document context pools
Generator	Qwen2.5-7B-Instruct; greedy; max input 4096; max new tokens 32
Candidate scorer input	$[q; S_k; c]$; max length 512; candidate budget 192; current answer excluded
Matched w/o-state input	$[q; c]$ only; all training rows/objectives/update budget/verifier/policy/calibration otherwise matched
Scorer initialization	Common shallow retrieved-state precursor, epoch 2; selected without clean dev or held-out data
Verifier input	$[q; S_k; a_k]$; training cap 384, deployment cap 512; answer budget 64
Verifier training	7k train / 1k dev questions; 21k / 3k unique states; BGE-M3 init; BCE; best checkpoint epoch 2
Maximum depth	5 documents
Candidate selection	$c^* = \arg\max_c b(q, S_k, c)$; retrieval-rank then pool-index tie break
Full stopping rule	$g_k = b_k^* - 4\,\mathrm{logit}(p_C)$ with frozen τ_2, τ_3, τ_4
Scorer selection	Mean of harm AP, benefit AP, harm-pair, benefit-pair on mixed dev
Calibration	Independent 1,000-question split; shared policy family; held-out questions excluded
Held-out evaluation	Frozen 2,000 questions; 1,260 2Wiki / 740 Hotpot
Paired significance	20,000 paired bootstrap + 20,000 sign-flip replicates

Reference answers and ΔV are never available to the deployed scorer or verifier at held-out inference. Counterfactual generator calls are training-only, and all method-selection and calibration choices are frozen before held-out reporting.