APPRAG: WHEN RELEVANT EVIDENCE DOES NOT APPLY IN RETRIEVAL-AUGMENTED GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

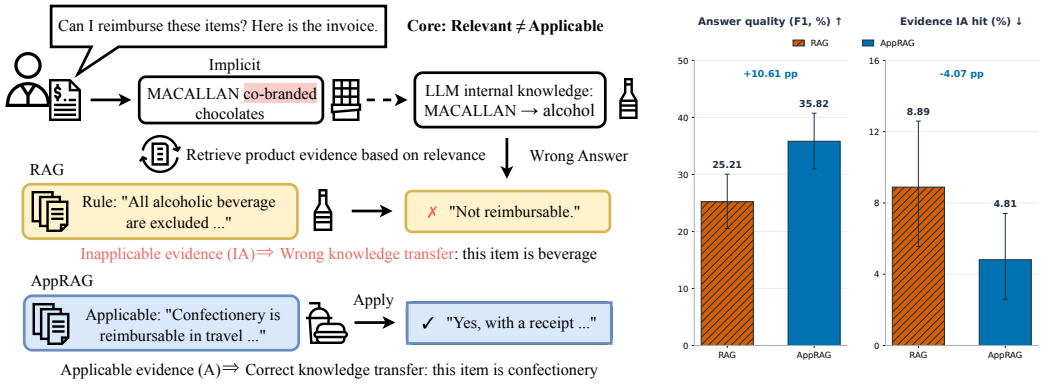
Retrieval-augmented generation (RAG) uses retrieved text passages as evidence to answer knowledge-intensive questions. However, relevant and factually correct evidence may not satisfy the conditions of the reasoning step it is used to support. When the context can include only a limited number of passages, applicable evidence may rank too low to be selected. Meanwhile, individually useful passages may not jointly support all necessary reasoning steps. To address these evidence-selection challenges, we introduce AppRAG, a three-stage framework for selecting applicable and complementary evidence. First, *requirement-conditioned applicability modeling* assesses whether evidence supports question-derived requirements under matching conditions, distinguishing valid partial support from condition mismatch while keeping unresolved evidence separate. Second, *applicability-guided embedding calibration* uses these assessments to update the query representation and rerank available candidates, verifying evidence replacements before updating the context. Third, *applicability-aware evidence selection* constructs complementary evidence sets under a per-context document budget and compares the answer drafts generated from each set. The answer generator remains frozen throughout. Our analysis characterizes when missing condition-match information imposes unavoidable evidence-selection loss. On evaluation sets from HotpotQA, MuSiQue, and 2WikiMultiHopQA, AppRAG improves Qwen3-8B's overall token-level F1 by 2.82–13.43 pp across all eight tested RAG integrations, including a 10.61-point gain over Vanilla RAG. Our code is available at `https://anonymous.4open.science/r/AppRAG-C2E8`.

1 INTRODUCTION

Retrieval-augmented generation (RAG) grounds large language models in external evidence for knowledge-intensive questions (Lewis et al., 2020; Zeng et al., 2026). Existing work improves candidate ordering (Sung et al., 2023), assesses evidence reliability (Asai et al., 2024), and selects passages to cover information requirements (Lee et al., 2025). However, these objectives do not by themselves establish whether evidence supports a reasoning step under matching conditions. A passage may state a correct fact about the right topic while violating the entity, relation, time, or scope conditions of its intended use. Figure 1 shows a real-world reimbursement case from a RAG system: MACALLAN co-branded chocolates are classified as confectionery, but their association with alcohol does not make the alcohol-exclusion rule applicable. The rule can be correct while its use is unjustified. We define *evidence applicability* as support for a question-specific requirement under matching conditions (Definition 1). Applicability is necessary for a fact to justify an inference; accurate sources alone do not ensure a valid conclusion. Under a limited context budget, mismatched evidence can displace valid support, while redundant passages leave reasoning steps uncovered. Partial support remains useful even when it does not answer the whole question. We characterize when missing condition-match information forces selection loss (Theorem 1).

Selecting applicable and complementary evidence under a limited context budget raises three challenges. **First, relevance can mask condition mismatch.** A passage may match the topic yet violate an entity, relation, time, or scope condition of its intended use, yielding relevant but inapplicable (IA) evidence. This differs from valid partial support: a bridge passage may justify an intermediate reasoning step without answering the whole question. Missing information also does not establish a mismatch, so unresolved evidence must remain distinct from confirmed inapplicability. **Second,**

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.



(a) Relevant evidence can still be inapplicable. **(b)** AppRAG improves over RAG.

Figure 1: **Relevance does not guarantee applicability.** (a) A real-world RAG QA example. (b) Answer F1 and confirmed IA hits for AppRAG and Vanilla RAG (RAG).

applicable evidence can remain below the context cutoff. Recognizing useful support does not ensure that it enters the selected context. Applicability assessments must influence candidate ordering, but promoting one passage may displace evidence needed for another requirement. The resulting context replacements therefore need to be checked rather than accepted from rank alone. **Third, individually applicable passages can leave the reasoning incomplete.** Several passages may support the same requirement while another necessary step remains unsupported. Selection must therefore account for complementary support across requirements rather than passage scores alone. Moreover, coverage alone does not ensure correct evidence use.

To address these challenges, we introduce AppRAG, a three-stage framework for selecting applicable and complementary evidence with a frozen answer generator. Requirement-conditioned applicability modeling learns applicability and coverage signals from question-derived requirements, distinguishing valid partial support, condition mismatch, and unresolved evidence. Applicability-guided embedding calibration uses these assessments in a learned residual query update to rerank candidates while keeping document embeddings fixed. An answer-blind verifier checks proposed context replacements. Applicability-aware evidence selection constructs complementary contexts under a per-context document budget and compares their context–answer pairs. An alternative is accepted only when comparison judgments agree and it passes a learned IA-increase risk gate.

Experiments across eight RAG methods show consistent improvements in dataset-averaged answer F1, supporting the use of applicability-guided selection with different retrieval strategies. Fixed-candidate analyses further show improved support-document ranking and benefits from verifying context updates. Overall, our contributions are:

- To distinguish valid support from mismatch, we formalize requirement-conditioned applicability and characterize selection loss from missing condition-match information.
- To promote under-ranked applicable evidence, we combine applicability-guided query calibration over fixed candidates with answer-blind verification of context exchanges.
- To address incomplete coverage, we construct complementary contexts under a document budget and select context–answer pairs using agreement and an IA-increase risk gate.

2 Related Work

Retrieval and evidence ranking. Limited candidate coverage and imperfect ordering motivate passage reranking (Pradeep et al., 2023; Zhang et al., 2025a), query adaptation (Sung et al., 2023), graph-based retrieval (Jiménez Gutiérrez et al., 2025; Guo et al., 2025), and reasoning-guided candidate expansion (Kim et al., 2026). Generator-aligned pruning further addresses evidence that is weak or harmful for generation (Song et al., 2026). High rank alone does not establish that a passage satisfies the conditions of its intended use. AppRAG uses requirement-conditioned applicability assessments to guide ranking updates and verifies the resulting context exchanges before acceptance.

Verification and evidence use. Unreliable retrieval and unsupported answers motivate evidence assessment and filtering (Asai et al., 2024; Yan et al., 2024; Saxena et al., 2026), structured reasoning checks (Zhao et al., 2026; Fang et al., 2026), and draft validation (Dhuliawala et al., 2024; Wang et al., 2025). Response refinement and interaction with context further improve evidence use (Madaan et al., 2023; Ye et al., 2026). Answer-level verification alone does not determine which passages remain applicable to intermediate reasoning steps. AppRAG distinguishes applicable partial support, established condition mismatch, and unresolved evidence, using these assessments to guide context construction before comparing answer drafts.

Conditions and context sufficiency. Answer validity can depend on conditions (Sun et al., 2022; Lin et al., 2024), while evidence sources can conflict (Jiayang et al., 2024) or collectively lack sufficient support (Joren et al., 2025; Qiu et al., 2026). Requirement-based selection (Lee et al., 2025; Unda, 2026) and sufficiency-guided retrieval (Li et al., 2026; Ha Lan N.T et al., 2026) address coverage and missing information. Applicability and context sufficiency address different questions: whether a passage supports a requirement under matching conditions, and whether the combined context provides enough support to answer. AppRAG builds complementary contexts using applicability-conditioned requirement coverage and compares their answer drafts.

3 METHOD

3.1 PROBLEM FORMULATION AND OVERVIEW

Let $\mathcal{R}(q) = \{r_j\}_{j=1}^{m}$ denote the facts and conditions elicited from question q by a frozen requirement generator before evidence assessment. These predicted requirements remain fixed during inference and need not match gold reasoning steps. A *document* is a retrieved passage with its title and text.

Definition 1 (Evidence applicability)**.** For passage d and requirement r_j, let $\mathrm{supp}(d, r_j)$ denote support for the proposed reasoning step and $\mathrm{match}(q, d, r_j)$ indicate compatibility with its entity, relation, time, and scope conditions. Define

$$\begin{aligned} \alpha_j(q,d) &= \mathbf{1}[\mathrm{supp}(d,r_j) = 1 \ \wedge \ \mathrm{match}(q,d,r_j) = 1], \\ \mathrm{App}(q,d) &= \max_{1 \le j \le m} \alpha_j(q,d). \end{aligned} \tag{1}$$

A passage is applicable when $\mathrm{App}(q, d) = 1$, even if it supports only one intermediate step. IA requires an established mismatch in a proposed use, so neither missing support nor $\mathrm{App}(q, d) = 0$ alone establishes IA. A valid use for one step does not preclude misuse for another.

Problem statement. Retrieval supplies an initial pool $\mathcal{C}_0(q)$ and a materialized candidate union $\mathcal{C}(q) \supseteq \mathcal{C}_0(q)$, constructed according to the retrieval interface (Appendix E.1). Given q, $\mathcal{R}(q)$, $\mathcal{C}(q)$, and a fixed generator G, we select an ordered context S and return $\hat{y}_S = G(q, S)$. We optimize over $\mathcal{S}_{B,T}(q)$, the family of contexts with at most B distinct documents and T evidence tokens:

$$\underset{S \in \mathcal{S}_{B,T}(q)}{\text{maximize}} \ V(q,S) \qquad \text{and} \qquad \underset{S \in \mathcal{S}_{B,T}(q)}{\text{minimize}} \ H(q,S), \tag{2}$$

where V counts requirements covered by applicable evidence, counting each requirement once across the context, and H is the fraction of IA documents.

From applicability to joint selection. Appendix B proves conditions for unavoidable selection loss from missing condition-match information and a tight budget-dependent bound under condition ambiguity. Even with exact applicability, deciding budgeted requirement coverage is NP-complete when B is part of the input (Theorem 3). These results characterize the global problem; AppRAG uses predicted applicability to construct a finite family of coverage-oriented contexts and compares their drafts. The implementation enforces a per-context document cap and checks serialized prompt length; candidate access and total inference computation are reported separately (Appendix E.2).

Overview. AppRAG proceeds through three stages (Figure 2). *Applicability modeling* (§3.2) uses question-derived requirements to score candidates and construct the initial context S_{app}. *Applicability-guided embedding calibration* (§3.3) updates the query representation and verifies proposed context exchanges without using answer drafts, producing S_{cal}. *Applicability-aware evidence selection* (§3.4) compares drafts from S_{cal} and coverage-oriented alternatives, then uses comparison agreement and an IA-increase risk gate to return $(S^*, \hat{y}_{S^*})$. § 3.5 describes training and inference.

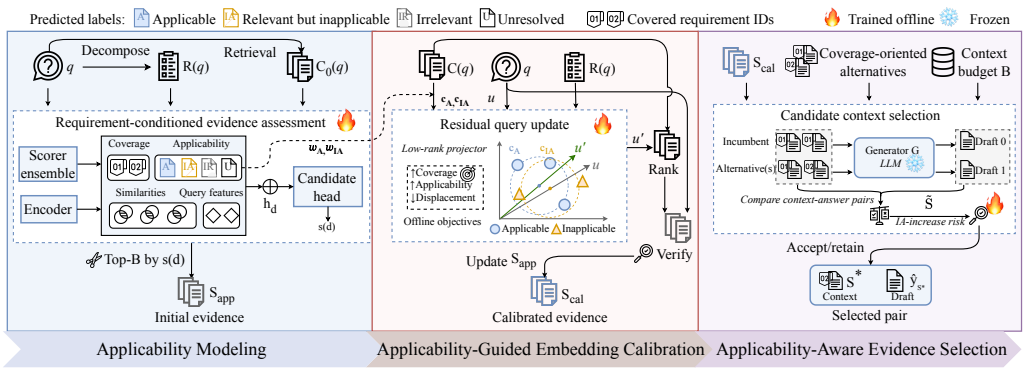


Figure 2: **AppRAG framework.** Evidence assessments guide query calibration and verified context updates. Coverage-oriented alternatives are constructed using cached verifier assessments and compared with the incumbent through their generated drafts. Coverage badges denote predicted covered requirement IDs; alternative construction uses the verifier's consensus IDs.

3.2 Applicability Modeling

Assessment semantics. The assessment pipeline estimates the semantic target in Definition 1. It distinguishes applicable evidence (A), relevant but inapplicable evidence (IA), irrelevant evidence (IR), and unresolved evidence (U). In particular, U is retained when the available information establishes neither support nor a condition mismatch. The support and condition-match predicates in Definition 1 specify the semantic target, not two independently verified model outputs.

Requirement-conditioned scoring. A scorer ensemble adapted with LoRA (Hu et al., 2022) supplies document-level applicability, class, support, and coverage predictions. A candidate head combines these predictions with question–document similarity, summary statistics of requirement–document similarities, and query-level requirement features:

$$\mathbf{h}_d = \Phi\big(q, \mathcal{R}(q), d; f_\theta\big), \qquad s(d) = \big[\sigma(g_\eta(\mathbf{h}_d))\big]_{\text{app}}, \qquad S_{\text{app}} = \text{Top}_B(\mathcal{C}_0; s). \tag{3}$$

Here f_θ is the scorer ensemble, Φ constructs candidate features and standardizes them using training statistics, g_η is the candidate head, and σ is the elementwise sigmoid. The applicability output $s(d)$ orders candidates within $\mathcal{C}_0$; Top_B returns the highest-scoring documents in score order. The first selected document is the exchange anchor a. The ensemble uses multitask supervision; the candidate head combines pointwise classification with within-question A-over-IA ranking (§ 3.5).

The same assessment pipeline scores documents in $\mathcal{C}$ for calibration. Its A weight $w_{\text{A}}(d)$ averages the ensemble's binary applicability prediction and A-class probability. Its IA weight $w_{\text{IA}}(d)$ is the calibrated residual IA score, obtained from the predicted IA probability and non-applicability. These are predicted document-level signals. Requirement conditioning enters through the scoring features; it does not supply a verified label for each requirement–document pair.

3.3 Applicability-Guided Embedding Calibration

To bring applicable evidence below the context cutoff into consideration, calibration updates the query representation using requirements and predicted evidence distributions across the candidate pool. The updated ranking proposes context exchanges, which an answer-blind verifier evaluates before acceptance. Appendix C.1 compares this policy with direct $s(d)$ ranking on the same pool.

Residual query update. Let $\mathbf{u}$ and $\mathbf{v}_d$ be normalized query and document embeddings, and $\bar{\mathbf{r}}$ the normalized mean requirement embedding. The applicability weights define two evidence centers:

$$\mathbf{c}_L = \text{norm}\left(\frac{\sum_{d\in\mathcal{C}} w_L(d)\mathbf{v}_d}{\sum_{d\in\mathcal{C}} w_L(d)}\right), \qquad L \in \{\text{A}, \text{IA}\}. \tag{4}$$

Here norm denotes ℓ_2 normalization; a center defaults to $\mathbf{u}$ when its weight mass does not exceed a numerical tolerance. A low-rank projector combines the original query with offsets for the require-

ments and the two evidence centers:

$$\mathbf{z} = [\mathbf{u}; \bar{\mathbf{r}} - \mathbf{u}; \mathbf{c}_{\mathrm{A}} - \mathbf{u}; \mathbf{u} - \mathbf{c}_{\mathrm{IA}}], \tag{5}$$

$$\mathbf{u}' = \mathrm{norm}(\mathbf{u} + W_{\mathrm{out}} \tanh(W_{\mathrm{in}}\mathbf{z} + \mathbf{b}_{\mathrm{in}})), \qquad t(d) = \mathbf{v}_d^{\mathsf{T}}\mathbf{u}'. \tag{6}$$

W_{in}, W_{out}, and $\mathbf{b}_{\mathrm{in}}$ are learned projection parameters, collectively denoted ω. Their training objective favors applicable and requirement-covering documents while penalizing query displacement (§ 3.5). Document embeddings remain fixed. The score $t(d)$ ranks the materialized union $\mathcal{C}$; the calibrated query does not initiate another corpus search.

Verified context update. Let K denote the candidate ranking depth within $\mathcal{C}$. The per-context document budget is B. We form an exchange envelope $E_K = \mathrm{Top}_K(\mathcal{C}; t) \cup \{a\}$ and start from S_{app}. An answer-blind verifier assigns each candidate an applicability label, confidence, conflict flag, and covered requirements from q, $\mathcal{R}(q)$, and the evidence. For a context S, define

$$\boldsymbol{\kappa}(S) = \big(N_{\mathrm{A}}(S),\ -N_{\mathrm{IA}}(S),\ -N_{\mathrm{conflict}}(S),\ \gamma_{\mathrm{A}}(S)\big), \tag{7}$$

where the counts refer to verifier predictions and γ_{A} sums confidence over A-labeled documents. Each step replaces a non-anchor document with an unselected candidate from E_K only if $\boldsymbol{\kappa}$ increases lexicographically. We choose the greatest resulting signature and use $s(d)$ to break ties, followed by document identifiers and slot indices. When no such exchange remains, the current context is S_{cal}. The anchor and context size stay fixed; if no exchange is accepted, $S_{\mathrm{cal}} = S_{\mathrm{app}}$. Improvement in $\boldsymbol{\kappa}$ reflects cached verifier predictions and does not guarantee better true applicability or answer quality.

3.4 Applicability-Aware Evidence Selection

Ranking documents individually can select redundant support and leave some requirements uncovered. Even a context covering all requirements may yield an incorrect answer. AppRAG therefore constructs coverage-oriented alternatives and compares their generated drafts.

Coverage-oriented candidate contexts. AppRAG forms a finite family $\mathcal{F}(q)$ containing S_{cal} and coverage-oriented alternatives, deduplicated by document order. Let $\widehat{\mathcal{J}}(d)$ contain the fixed requirement IDs that both verifier views judge passage d to support under the question's conditions; this set is empty unless d is labeled A and conflict-free. We measure context coverage by the number of distinct requirements supported according to these judgments: $\widehat{V}(q, S) = |\bigcup_{d \in S} \widehat{\mathcal{J}}(d)|$, counting each requirement once. Each alternative starts empty and greedily selects from a calibrated or fused top-K pool the conflict-free A document adding the most uncovered requirements, breaking ties by verifier confidence and candidate order. Alternatives use compact or full document budgets without a fixed anchor. When no candidate adds coverage, fallback rules fill the remaining slots from the same pool (Appendix E.3). Appendix E details verifier consensus and ranking.

Draft comparison. The generator produces $\hat{y}_S = G(q, S)$ for each $S \in \mathcal{F}(q)$. A frozen judge evaluates the context–answer pairs using the question, requirements, evidence, and verifier assessments, retaining the incumbent unless an alternative is clearly preferable. Only the alternatives are anonymously coded and shown in opposite orders. If both judgments validly select the same context, it becomes the proposal $\widetilde{S}$; otherwise, $\widetilde{S} = S_{\mathrm{cal}}$.

IA-increase risk gate. A logistic head $\rho_\psi(\mathbf{x})$ estimates whether the proposal increases the fraction of IA documents. Its standardized evidence/action features $\mathbf{x}$ describe the proposal and its differences from S_{cal}, excluding draft text and reference answers. With a fixed threshold τ,

$$S^* = \begin{cases} \widetilde{S}, & \widetilde{S} \neq S_{\mathrm{cal}} \text{ and } \rho_\psi(\mathbf{x}) < \tau, \\ S_{\mathrm{cal}}, & \text{otherwise}, \end{cases} \qquad \hat{y} = \hat{y}_{S^*}. \tag{8}$$

Here $\hat{y}$ abbreviates the selected draft $\hat{y}_{S^*}$. Core AppRAG returns this saved draft without another generation step; the ACE-RAG integration adds playbook-conditioned generation (Appendix E.1). The agreement rule resolves inconsistent comparisons, while the risk gate checks whether a proposed change is predicted to increase the IA document fraction. Appendix E.3 reports the gate's fitting procedure and its effect under fixed proposals.

3.5 Training and Inference

Offline training. AppRAG separately fits the scorer adapters and heads, candidate head, query projector, and action-risk head on development questions; pretrained backbone weights stay fixed. Supervision combines applicability and requirement-coverage annotations with official support labels, excluding main-table evaluation questions. The query projector connects applicability predictions to candidate ranking through three terms:

$$\mathcal{L}_{\text{proj}} = \mathcal{L}_{\text{rank}} + \mathcal{L}_{\text{cov}}^{\text{proj}} + \lambda_{\text{disp}}\mathbb{E}[1 - \mathbf{u}'^{\mathsf{T}}\mathbf{u}], \tag{9}$$

where $\mathcal{L}_{\text{rank}} = \mathcal{L}_{\text{list}} + \mathcal{L}_{\text{sep}}$ prioritizes applicable documents, while $\mathcal{L}_{\text{cov}}^{\text{proj}}$ promotes requirement-level support. Applicable passages can cover the same requirement, so the additive form combines document-level ranking with requirement-specific score margins in a single projector update. Since $\mathbf{u}$ and $\mathbf{u}'$ have unit norm, $1 - \mathbf{u}'^{\mathsf{T}}\mathbf{u} = \frac{1}{2}\|\mathbf{u}' - \mathbf{u}\|_2^2$; the final term therefore imposes a quadratic penalty on query movement within the fixed document embedding space. The coefficient λ_{disp} controls the trade-off between the ranking losses and displacement. We use unit weights for the ranking and coverage terms and $\lambda_{\text{disp}} = 0.1$.

Frozen inference. All fitted parameters and normalization statistics are fixed at inference. For each question, AppRAG derives requirements, scores evidence, verifies proposed exchanges, and compares drafts without access to reference answers or evaluation labels. The answer generator, encoders, verifier, and comparison judge receive no parameter updates. Algorithms 1 and 2 give the full procedures; Appendix D.3 analyzes their cost.

4 Experiments

4.1 Settings

Datasets and evaluation. We evaluate AppRAG on fixed evaluation sets of HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), and 2WikiMultiHopQA (2Wiki) (Ho et al., 2020). All methods use the same questions within each dataset. We report exact match (EM) and token-level F1 as percentages. Overall is the unweighted mean across datasets; ΔF1 is the signed percentage-point (pp) change from the corresponding baseline. Appendix A details data sources, sampling, and statistics; Appendix E covers scoring and indexes supplementary evaluation protocols.

Baselines. We compare LLM-only and Vanilla RAG with EntityLinkRAG, our title-linking and lexical-retrieval baseline related to Granata et al. (2026); HippoRAG2 (Jiménez Gutiérrez et al., 2025); LightRAG (Guo et al., 2025); and ACE-RAG, our QA adaptation of agentic context engineering (Zhang et al., 2026). We also include RRF-ensemble and pipelines using RankZephyr-7B (Pradeep et al., 2023) or REARANK-7B (Zhang et al., 2025a), denoted RankZephyr-RAG and REARANK-RAG, respectively. We evaluate each RAG system with and without AppRAG. Standalone AppRAG uses Vanilla RAG; baseline adaptations are detailed in Appendix E.1.

Models and per-context document budget. Qwen3-8B (Yang et al., 2025) generates answers and candidate drafts with greedy decoding; a learned Qwen3-Reranker-4B scorer ensemble assesses evidence, and Qwen3-32B serves as the answer-blind verifier and context–answer comparison judge. All parameters remain frozen at inference. Each context contains at most $B = 4$ documents; token lengths, cumulative evidence access, and total compute vary across systems (Appendix E.2).

Retrieval protocol. Graph baselines index the union of BGE-M3's top-four passages across sampled questions per dataset, yielding prefiltered corpora. Each integration uses its specified retrieval interface and candidate boundary; ACE-RAG shares a development-frozen playbook across both conditions. Appendix E.1 details corpus access and context construction.

4.2 Main Results

Table 1 shows overall F1 gains of 2.82–13.43 percentage points across all eight integrations, including an increase from 38.54 to 45.30 for ACE-RAG, the strongest baseline. These results support AppRAG as a complementary evidence-selection layer across retrieval pipelines. HippoRAG2 and LightRAG outperform Vanilla RAG but remain below standalone AppRAG after augmentation; these comparisons use prefiltered graph indexes and do not isolate graph structure. Gains are not

Table 1: Multi-hop QA with Qwen3-8B (EM/F1, %). Bold: pairwise best; darker blue: higher values within each column; ΔF1: overall gain (pp).

Method	HotpotQA		MuSiQue		2Wiki		Overall		
	EM	F1	EM	F1	EM	F1	EM	F1	ΔF1
Base Models and Vanilla RAG									
LLM-only	23.33	32.78	4.44	16.63	38.89	40.48	22.22	29.97	–
Vanilla RAG (2020)	40.00	44.25	15.56	18.19	12.22	13.17	22.59	25.21	–
+ **AppRAG**	**55.56**	**63.96**	**17.78**	**24.67**	**17.78**	**18.84**	**30.37**	**35.82**	**+10.61**
Graph-based and Advanced RAG									
HippoRAG2 (2025)	37.78	44.81	15.56	20.23	13.33	14.39	22.22	26.48	–
+ **AppRAG**	**44.44**	**52.42**	**17.78**	**23.21**	**16.67**	**18.28**	**26.30**	**31.30**	**+4.82**
LightRAG (2025)	**44.44**	50.31	14.44	19.35	12.22	13.17	23.70	27.61	–
+ **AppRAG**	**44.44**	**51.59**	**16.67**	**21.42**	**16.67**	**18.28**	**25.93**	**30.43**	**+2.82**
EntityLinkRAG† (2026)	35.56	38.72	5.56	9.58	10.00	10.95	17.04	19.75	–
+ **AppRAG**	**54.44**	**61.67**	**17.78**	**22.21**	**14.44**	**15.65**	**28.89**	**33.18**	**+13.43**
ACE-RAG† (2026)	50.00	57.80	25.56	32.00	21.11	25.83	32.22	38.54	–
+ **AppRAG**	**56.67**	**62.78**	**28.89**	**35.76**	**33.33**	**37.37**	**39.63**	**45.30**	**+6.76**
Reranking-enhanced RAG									
RRF-ensemble‡ (2009)	47.78	53.32	15.56	19.61	13.33	14.21	25.56	29.04	–
+ **AppRAG**	**55.56**	**63.96**	**17.78**	**24.67**	**17.78**	**18.84**	**30.37**	**35.82**	**+6.78**
RankZephyr-RAG (2023)	44.44	52.30	**20.00**	**25.92**	12.22	12.54	25.56	30.25	–
+ **AppRAG**	**55.56**	**63.94**	18.89	25.33	**17.78**	**18.84**	**30.74**	**36.04**	**+5.79**
REARANK-RAG (2025a)	45.56	53.01	**21.11**	25.08	13.33	15.66	26.67	31.25	–
+ **AppRAG**	**55.56**	**63.39**	18.89	**25.33**	**17.78**	**18.84**	**30.74**	**35.85**	**+4.60**

† Adaptations of ELERAG and ACE, respectively. ‡ RRF-ensemble: RRF over three rerankers.

uniform: on MuSiQue, RankZephyr-RAG loses F1, while REARANK-RAG loses EM despite a small F1 increase. On 2Wiki, LLM-only still achieves higher F1 than all tested retrieval configurations. AppRAG therefore improves overall performance over paired retrieval baselines without eliminating local regressions or the gap to LLM-only answering on this dataset.

4.3 Ablation Analysis

Overall Analysis. We ablate each stage in Figure 2 while keeping the remaining learned components fixed. Table 2 reports relative changes from full AppRAG with Qwen3-4B, using scores averaged equally across datasets. Removing Modeling causes the largest degradation across all four metrics, making it the main contributor in this configuration. Removing Selection reduces F1 more than removing Calibration despite similar coverage losses, suggesting that coverage alone does not explain answer quality. Cov and IA-risk are lower-bound diagnostics of requirement coverage and answer-level applicability mismatches, respectively. They remain descriptive rather than validated semantic measures because the LLM judge failed prespecified quality checks (Appendix E.4).

Table 2: Module ablation (relative %).

Configuration	EM↑	F1↑	Cov↑	IA-risk↓
w/o Modeling	↓10.81%	↓11.26%	↓9.56%	↑7.46%
w/o Calibration	↓1.87%	↓1.58%	↓1.79%	↑3.54%
w/o Selection	↓4.78%	↓5.10%	↓1.72%	↑2.61%

Context replacement verification. With the verifier's other uses unchanged, retaining replacement verification improves token-level F1 and semantic accuracy by 5.36 and 5.00 pp, respectively (Table 3), with paired 95% bootstrap intervals above zero. These gains support checking context updates as a complement to calibration's ranking improvements, but do not establish the accuracy of the verifier's labels. Appendix E.5 details the paired evaluation protocol and semantic scoring.

Table 3: Replacement verification (%).

Configuration	F1↑	Acc.↑
w/o verification	29.73	33.89
AppRAG	**35.09**	**38.89**
Gain (pp)	+5.36	+5.00
95% CI (pp)	[1.47, 9.33]	[1.11, 9.44]

We evaluate query calibration against official support-document annotations, using identical candidate pools and document embeddings before replacement verification. Figure 3 shows gains of 3.15 and 3.38 percentage points in Recall@4 and Recall@10, respectively, with positive paired 95% bootstrap intervals. Complete-support coverage@10, the fraction of questions whose top ten documents contain every annotated support document, rises from 45.00% to 50.56%. Calibration therefore improves both support-document recall and recovery of the complete annotated support set. Because the candidate pool and document embeddings remain fixed, these gains reflect better ordering of available evidence. Appendix E.5 details the protocol; Appendix C.4 reports applicability recall across depths and datasets.

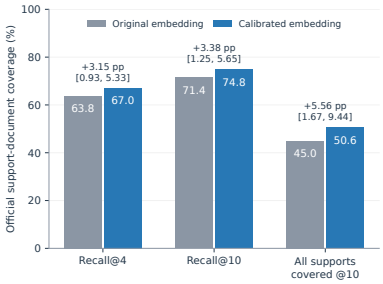


Figure 3: Query calibration gains.

Marginal Benefit Analysis. AppRAG improves observed semantic accuracy and token-level F1 across Qwen3 sizes from 0.6B to 32B (Figure 4). Gains peak at intermediate sizes and remain positive at 32B, although adjusted adjacent-size comparisons do not establish a stepwise decline. Selection contributes less at 32B, while removing Modeling or Calibration reduces accuracy at medium and large sizes but not at the smallest scale, indicating capacity-dependent benefits. Panels B and C assess sequential additions and individual removals without retraining, respectively; their effects are configuration-dependent. All sizes share a short-answer protocol evaluated separately from the preceding table (Appendix E.5). Markers show midpoints; shading reflects judgment uncertainty, and error bars include sampling uncertainty.

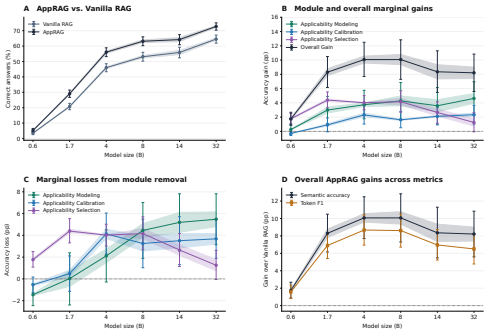


Figure 4: AppRAG benefits across scales.

Complementary experiments and design controls. Orthogonal experiments with two embedding models show improvements in mean support-document recall, ranking quality, and downstream answer quality, with larger retrieval gains under tighter context budgets (Appendix C.5). A separate study finds positive mean F1 and EM gains with all three tested rerankers, supporting compatibility across relevance-ranking components, although the smallest generator benefits little (Appendix C.6). These studies use different samples, so their gain magnitudes are not directly comparable. On a separate retrospective panel, calibrated ranking with verification improves F1 by 2.52 points over raw ranking with the same verification, supporting calibration's contribution when context updates are checked. The full procedure, using multiple drafts and online checks, attains the highest F1 among the tested policies. Controls for document scoring, training objectives, and final comparison rules further support the combined procedure, but do not isolate gains from its specific training loss or final objective (Appendices C.1–C.3).

4.4 IA Exposure in Selected Contexts

We examine whether final contexts contain relevant but inapplicable evidence. Figure 5 reports question-level IA-hit rates: an entity-level hit requires a confirmed entity mismatch, while an evidence-level hit requires a passage judged inapplicable to the question's requirements. AppRAG has lower observed rates at both levels across all three datasets, suggesting that its selected contexts expose the generator to fewer detected applicability mismatches. MuSiQue shows the largest evidence-level reduction. The reductions on 2Wiki are smaller, and both paired 95% bootstrap intervals touch zero, leaving their statistical significance unresolved. Violins summarize 10,000 paired question-bootstrap resamples; orange labels report AppRAG minus Vanilla RAG in percentage points. Because context lengths can differ, these results measure exposure in the final contexts rather than selection quality at a fixed length. Lower exposure also does not establish whether generated answers avoid adopting mismatched evidence. Appendix E.5 details the evaluation.

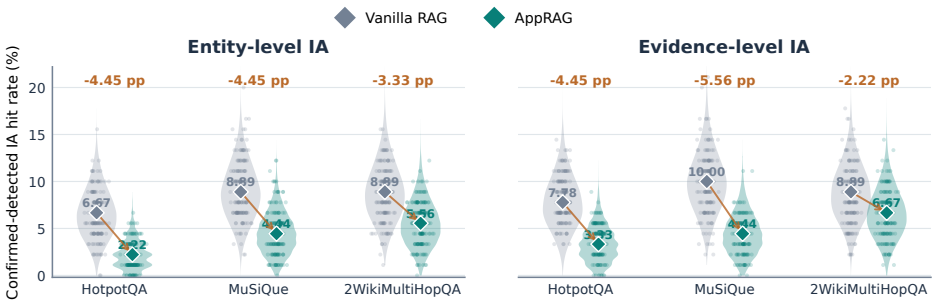


Figure 5: IA-hit rates (↓). Diamonds: observed rates; violins: paired bootstrap distributions.

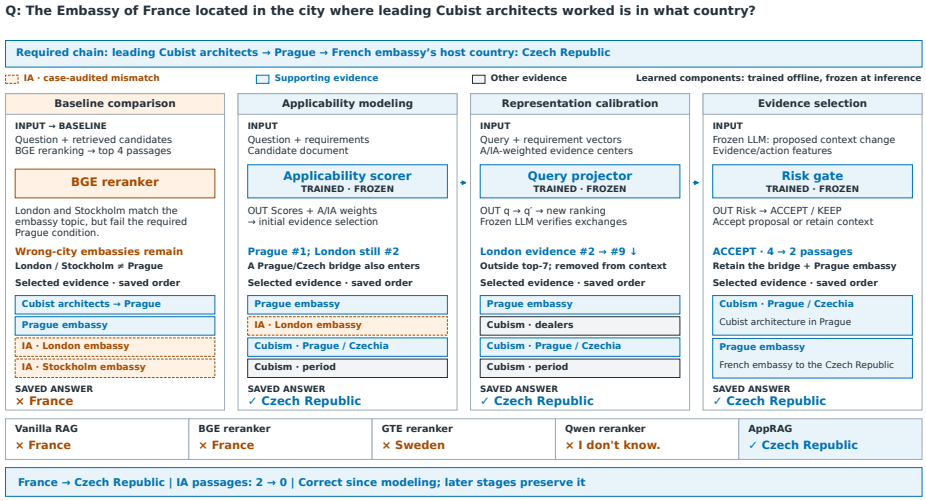


Figure 6: Filtering wrong-city evidence on MuSiQue while preserving a correct answer. Blue and orange indicate support and condition mismatches, respectively, based on case-level inspection.

5 Case Study

Filtering evidence while preserving the answer. The question asks for the host country of the French embassy in the city associated with leading Cubist architects (Figure 6). The BGE-reranked context includes Prague evidence alongside London and Stockholm embassy passages that fail the city condition, and yields France. AppRAG's initial context S_{app} includes the Prague embassy and a Cubist-architecture bridge to Prague/Czechia; its draft already answers Czech Republic despite retaining London. Calibration moves the London passage from second in the initial applicability ordering to ninth in the union ranking, outside the top-seven envelope and S_{cal}. Final selection retains the complementary bridge and embassy passages in S^*, preserving the answer with two passages. Relative to BGE, the final context reduces audited inapplicable passages from two to zero. This trace illustrates evidence filtering and compaction after the answer is already correct. Appendix C.13 presents a missing-cast-evidence case, the shared setup, and inspection details.

6 Conclusion

AppRAG uses question-derived requirements to connect evidence applicability, ranking calibration, and context–answer selection. Applicability information guides verified exchanges and coverage-oriented context construction. Experiments on HotpotQA, MuSiQue, and 2Wiki, overall F1 improves across all eight tested integrations without updating the answer generator. Fixed-candidate diagnostics show better support-document ranking, while a paired ablation supports verifying context replacements. Together, these findings support using applicability beyond document scoring to construct and compare evidence sets under the evaluated conditions.

Ethics Statement

Experiments use public QA benchmarks. Generated answers can still contain unsupported claims or misuse retrieved evidence. Applicability predictions should therefore support, rather than replace, human judgment when errors have material consequences.

Reproducibility Statement

Appendices A and E document dataset sampling, training, model configurations, and evaluation protocols. Reproduction requires the specified backbones and fitted scorer, candidate-head, projector, and risk-head parameters.

AI Use Statement

Large language models (LLMs) were used solely for language editing to improve the manuscript's grammar, clarity, and style. They were not involved in methodological design, experimental implementation, or the interpretation of results, and did not contribute any original research content.

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. URL https://openreview.net/forum?id=hSyW5go0v8.

David Blackwell. Equivalent comparisons of experiments. *The Annals of Mathematical Statistics*, 24(2):265–272, 1953. doi: 10.1214/aoms/1177729032. URL https://doi.org/10.1214/aoms/1177729032.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2318–2335, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.137. URL https://aclanthology.org/2024.findings-acl.137/.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759, 2009. doi: 10.1145/1571941.1572114. URL https://cormack.uwaterloo.ca/cormacksigir09-rrf.pdf.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3563–3578. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.212. URL https://aclanthology.org/2024.findings-acl.212/.

Jiwei Fang, Bin Zhang, Changwei Wang, Jin Wan, and Zhiwei Xu. Graph of verification: Structured verification of LLM reasoning with directed acyclic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30665–30672, 2026. doi: 10.1609/aaai.v40i36.40322. URL https://ojs.aaai.org/index.php/AAAI/article/view/40322.

Francesco Granata, Francesco Poggi, and Misael Mongiovì. Enhancing retrieval-augmented generation with entity linking for educational platforms. *Big Data and Cognitive Computing*, 10(4):120, 2026. doi: 10.3390/bdcc10040120. URL https://doi.org/10.3390/bdcc10040120.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.568. URL https://aclanthology.org/2025.findings-emnlp.568/.

Ha Lan N.T, Minh-Anh Nguyen, and Dung D. Le. Latent abstraction for retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2604.17866.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Yufang Hou, Alessandra Pascale, Javier Carnerero-Cano, Tigran Tchrakian, Radu Marinescu, Elizabeth Daly, Inkit Padhi, and Prasanna Sattigeri. WikiContradict: A benchmark for evaluating LLMs on real-world knowledge conflicts from wikipedia. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 109701–109747. Curran Associates, Inc., 2024. doi: 10.52202/079017-3481. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c63819755591ea972f8570beffca6b1b-Paper-Datasets_and_Benchmarks_Track.pdf.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Cheng Jiayang, Chunkit Chan, Qianqian Zhuang, Lin Qiu, Tianhang Zhang, Tengxiao Liu, Yangqiu Song, Yue Zhang, Pengfei Liu, and Zheng Zhang. ECON: On the detection and resolution of evidence conflicts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7816–7844, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.447. URL https://aclanthology.org/2024.emnlp-main.447/.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515. PMLR, 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Jjr2Odj8DJ.

Richard M. Karp. *Reducibility among Combinatorial Problems*, pp. 85–103. Springer US, Boston, MA, 1972. ISBN 978-1-4684-2001-2. doi: 10.1007/978-1-4684-2001-2_9. URL https://doi.org/10.1007/978-1-4684-2001-2_9.

Jongho Kim, Jaeyoung Kim, Jihyuk Kim, Yu Jin Kim, Seung-won Hwang, and Moontae Lee. Adaptive retrieval for reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37366–37381. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1734. URL https://aclanthology.org/2026.acl-long.1734/.

Dahyun Lee, Yongrae Jo, Haeju Park, and Moontae Lee. Shifting from ranking to set selection for retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17606–17619, Vienna, Austria, July

2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.861. URL https://aclanthology.org/2025.acl-long.861/.

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

Minghan Li, Junjie Zou, Xinxuan Lv, Chao Zhang, and Guodong Zhou. S2G-RAG: Structured sufficiency and gap judging for iterative retrieval-augmented QA. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25846–25862. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1185. URL https://aclanthology.org/2026.acl-long.1185/.

Jiuheng Lin, Yuxuan Lai, and Yansong Feng. Chain of condition: Construct, verify and solve conditions for conditional question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16596–16611. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.968. URL https://aclanthology.org/2024.findings-emnlp.968/.

Zheng Liu, Chaofan Li, Shitao Xiao, Yingxia Shao, and Defu Lian. Llama2Vec: Unsupervised adaptation of large language models for dense retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3490–3500, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.191. URL https://aclanthology.org/2024.acl-long.191/.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023. doi: 10.52202/075280-2019. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10862–10878, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.585. URL https://aclanthology.org/2024.acl-long.585/.

Thinh Pham, Nguyen Phan Nguyen, Pratibha Zunjare, Weiyuan Chen, Yu-Min Tseng, and Tu Vu. SealQA: Raising the bar for reasoning in search-augmented language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zWb7ueH16c.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. RankZephyr: Effective and robust zero-shot listwise reranking is a breeze!, 2023. URL https://arxiv.org/abs/2312.02724.

Jingxi Qiu, Zeyu Han, and Cheng Huang. SURE-RAG: Sufficiency and uncertainty-aware evidence verification for selective retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2605.03534.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Yash Saxena, Ankur Padia, Mandar Chaudhary, Kalpa Gunaratna, Srinivasan Parthasarathy, and Manas Gaur. Ranking free RAG: Replacing re-ranking with selection in RAG for sensitive domains. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=O88FCPAPAj.

Zhipeng Song, Yizhi Zhou, Xiangyu Kong, Jiulong Jiao, Xinrui Bao, Xu You, Xueqing Shi, Yuhang Zhou, and Heng Qi. Less is more for RAG: Information gain pruning for generator-aligned reranking and evidence selection, 2026. URL https://arxiv.org/abs/2601.17532.

Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and Yu Cheng. ConflictBank: A benchmark for evaluating the influence of knowledge conflicts in LLMs. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 103242–103268. Curran Associates, Inc., 2024. doi: 10.52202/079017-3280. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/baf4b960d118f838ad0b2c08247a9ebe-Paper-Datasets_and_Benchmarks_Track.pdf.

Haitian Sun, William Cohen, and Ruslan Salakhutdinov. ConditionalQA: A complex reading comprehension dataset with conditional answers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3627–3637. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.253. URL https://aclanthology.org/2022.acl-long.253/.

Mujeen Sung, Jungsoo Park, Jaewoo Kang, Danqi Chen, and Jinhyuk Lee. Optimizing test-time query representations for dense retrieval. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 5731–5746. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.354. URL https://aclanthology.org/2023.findings-acl.354/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Victor P. Unda. Controllable evidence selection in retrieval-augmented question answering via deterministic utility gating, 2026. URL https://arxiv.org/abs/2603.18011.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. FreshLLMs: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13697–13720, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.813. URL https://aclanthology.org/2024.findings-acl.813/.

Zilong Wang, Zifeng Wang, Long Le, Huaixiu Steven Zheng, Swaroop Mishra, Vincent Perot, Yuwei Zhang, Anush Mattapalli, Ankur Taly, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. Speculative RAG: Enhancing retrieval augmented generation through drafting. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xgQfWbV6Ey.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation, 2024. URL https://arxiv.org/abs/2401.15884.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Fuda Ye, Jiachuan Wang, Yongqi Zhang, Lei Chen, and Shuangyin Li. BubbleRAG: Interactive cognitive offloading with thought bubble in retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 43540–43558. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.2163. URL https://aclanthology.org/2026.findings-acl.2163/.

Xingyuan Zeng, Zuohan Wu, Yue Wang, Chen Zhang, Quanming Yao, Libin Zheng, and Jian Yin. Da-rag: Dynamic attributed community search for retrieval-augmented generation. In *Proceedings of the ACM Web Conference 2026*, WWW '26, pp. 2195–2206, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400723070. doi: 10.1145/3774904.3792430. URL https://doi.org/10.1145/3774904.3792430.

Le Zhang, Bo Wang, Xipeng Qiu, Siva Reddy, and Aishwarya Agrawal. REARANK: Reasoning reranking agent via reinforcement learning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2458–2471, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.125. URL https://aclanthology.org/2025.emnlp-main.125/.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eC4ygDs02R.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1393–1412, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.103. URL https://aclanthology.org/2024.emnlp-industry.103/.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025b. URL https://arxiv.org/abs/2506.05176.

Wenxin Zhao, Jiachuan Wang, Yongqi Zhang, Shuangyin Li, Cheng Deng, Jun Wang, and Lei Chen. VoG: Enhancing LLM reasoning through stepwise verification on knowledge graphs. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0RdAmwfVku.

APPENDIX

Organization. Appendix A describes the datasets and sampling. Appendix B presents the theoretical analysis, and Appendix C provides supplementary experiments and case studies. Appendices D and E detail training, inference, implementation, and evaluation protocols. Appendix F summarizes the prompts, and Appendix G discusses limitations. Table 4 indexes the experimental analyses in the main text and appendices.

Table 4: Guide to the experimental analyses.

Location	Research question
Main-text evaluation	
§4.2	How does AppRAG perform across RAG pipelines?
§4.3	What does each module contribute?
§4.3	How do gains vary with generator size?
§4.3	How do query calibration and verified context exchange affect retrieval and answer quality?
§4.4	Does AppRAG reduce detected IA exposure in selected contexts?
Supplementary analyses	
C.1	How should applicability scores guide evidence selection?
C.2	Do automatic A/IR labels agree with human valid-support judgments?
C.3	How do training objectives affect answer F1?
C.4	Does query calibration improve applicability recall?
C.5, C.6	How does AppRAG perform across embedding models and rerankers?
C.7	How do candidate depth K and context budget B affect answer F1?
C.8	How does evidence selection compare with answer refinement?
C.9	How does AppRAG compare with closely related methods?
C.10	What does applicability contribute beyond requirement-based set selection?
C.11	How does the context budget affect online LLM token cost?
C.12	How does answer-invariant risk gating affect answer quality and IA exposure?
Qualitative analysis	
§5, C.13	How do evidence revisions change contexts and answers?

A DATASETS

A.1 DATASET DESCRIPTIONS

HotpotQA (Yang et al., 2018) combines Wikipedia facts through bridge questions about intermediate entities and comparison questions about shared properties. Its distractor setting supplies supporting articles, distractors, and sentence-level support annotations. **MuSiQue** (Trivedi et al., 2022) composes two- to four-hop reasoning chains while filtering shortcuts. MuSiQue-Ans is answerable; MuSiQue-Full, our source, also includes unanswerable contrasts. **2WikiMultiHopQA** (2Wiki) (Ho et al., 2020) combines Wikipedia text and Wikidata relations through templates and logical rules, with supporting facts and entity–relation paths. These datasets annotate multi-hop support, not condition mismatch: an unannotated passage is not automatically IA.

A.2 DATASET SOURCES AND SAMPLING

The main comparison uses questions from HotpotQA, MuSiQue, and 2Wiki. We use a frozen census that excluded prior study populations known at its creation, grouping questions by alphanumeric-normalized identity. For each dataset, a deterministic procedure with three fixed seeds selects three disjoint groups of questions. Identities are fixed before retrieval and generation, using no outputs, scores, or IA filtering. All methods retain the same questions without replacement after retrieval or answer failures. The seeds define question groups, not repeated stochastic generations.

All four components are fitted before evaluation and frozen. Main-table question IDs are absent from scorer, candidate-head, projector, and risk-head fitting populations; normalized question-text checks also find no overlap with scorer, candidate-head, or risk-head records. These checks concern

Table 5: Coarse IA* prevalence across six datasets.

Dataset	Sample size		Evidence (%)	Query (%)	
	Queries	Pairs	IA*	IA* hit	A/IA* coexistence
ConflictBank† (2024)	200	800	8.75	29.00	29.00
ECON† (2024)	200	446	22.42	40.50	32.00
FreshQA (2024)	574	5,829	50.42	94.77	75.44
RAGTruth (2024)	155	465	23.87	49.68	44.52
SEALQA (2026)	488	9,025	44.31	97.54	79.30
WikiContradict (2024)	54	108	6.48	12.96	12.96

Evidence rates use annotated query–evidence pairs; query rates measure IA* occurrence and A/IA* coexistence. †Sampled query subsets.

component training in this retrospective evaluation. Supplementary analyses use separately specified populations and are not pooled with the main comparison.

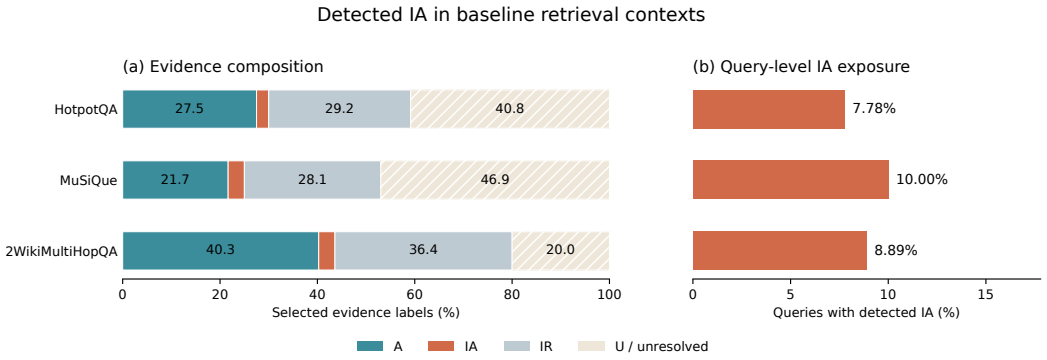


Figure 7: Detected IA in Vanilla RAG contexts. (a) Evidence-level A/IA/IR/U composition. (b) Percentage of queries containing at least one detected IA document.

A.3 Preliminary Evidence–Query Applicability Analysis

Table 5 summarizes a preliminary annotation study across six sources. Its IA* category, *relevant but judged not applicable*, includes relevant background and is broader than the explicit condition mismatch defining IA in § 3.2. We annotate query–evidence pairs using GPT-5.3 via its API, without reference answers. We retain nonempty candidate pools, excluding 26 FreshQA and 20 SEALQA empty pools. Evidence-level rates divide IA* pairs by all annotated pairs; query-level rates count questions with at least one IA* pair. All sources contain IA* evidence and pools where A and IA* coexist. These sampled collections motivate applicability analysis across settings; they do not estimate strict IA prevalence in general QA. The stricter three-benchmark audit appears in Figure 7.

B Theoretical Analysis of Applicability and Evidence Selection

Budgeted evidence selection requires both information about valid uses and a decision about which passages to combine. We analyze these two requirements separately. First, the support–match definition yields an exact correction for coverage lost through condition mismatch. We then characterize when discarding matching information creates a strict optimality gap and derive a tight budget-dependent bound under condition ambiguity. Finally, we show that joint selection remains NP-hard with perfect applicability information when the budget varies with the input. Together, these results connect applicability assessment to the subsequent selection problem studied by AppRAG.

B.1 A Bicriteria Selection Problem

Let $\mathcal{C}(q) = \{d_i\}_{i=1}^{n}$ and $\mathcal{R}(q) = \{r_j\}_{j=1}^{m}$, with $m \geq 1$. For full-information selection, the binary applicability table $\alpha_{ji} = \alpha_j(q, d_i)$ from Definition 1 is supplied as input. Let $\iota_i \in \{0, 1\}$ indicate an established IA judgment for the proposed use of d_i, and let ℓ_i be its positive integer token cost, including its document formatting. The evidence-token allowance T excludes the fixed question and prompt overhead. All numeric inputs have finite binary encodings. This is an explicit-table abstraction; the theorem does not require an algorithm for deciding natural-language applicability.

Writing $d_i \in S$ for membership regardless of order, define

$$\begin{aligned} V(q, S) &= \sum_{j=1}^{m} \max_{d_i \in S} \alpha_{ji}, \\ H(q, S) &= \frac{\sum_{d_i \in S} \iota_i}{\max\{1, |S|\}}, \qquad \mathcal{S}_{B,T}(q) = \left\{ S : |S| \leq B,\ \sum_{d_i \in S} \ell_i \leq T \right\}. \end{aligned} \tag{10}$$

Contexts contain distinct candidates, and the maximum over an empty context is defined as zero. Thus $V(q, \emptyset) = H(q, \emptyset) = 0$. Coverage counts each requirement once, even if several documents support it. IA exposure is a document fraction, consistent with the quantity targeted by the action-risk head. It is distinct from the answer-level IA-risk metric. Neither full coverage nor zero IA exposure guarantees that the generator answers correctly.

Equation 2 is interpreted in the Pareto sense: a feasible context dominates another if it has at least as much coverage and no greater IA exposure, with one strict inequality. A context is Pareto optimal when no feasible context dominates it. The objectives here depend on membership; document order can still affect $G(q, S)$. Coverage is credited only when a selected passage supports the atomic requirement under matching conditions. Support that can only be established jointly by several passages is not captured by this simplified coverage model.

Definition 2 (Budgeted applicability-aware selection)**.** Given the tables and costs above, budgets B, T, a coverage threshold $c \in \{0, \ldots, m\}$, and a rational exposure bound $\varepsilon \in [0, 1]$, decide whether there exists a context $S \in \mathcal{S}_{B,T}(q)$ such that

$$V(q, S) \geq c, \qquad H(q, S) \leq \varepsilon. \tag{11}$$

B.2 Applicability as a Correction to Coverage

Write the predicates in Definition 1 as binary values

$$b_{ji} = \mathbf{1}[\mathrm{supp}(d_i, r_j) = 1], \qquad c_{ji} = \mathbf{1}[\mathrm{match}(q, d_i, r_j) = 1], \qquad \alpha_{ji} = b_{ji} c_{ji}. \tag{12}$$

Here b_{ji} describes support for the proposed step before applying the separate condition-match test. It is a semantic predicate, rather than an arbitrary retrieval score. Suppressing q for a fixed instance, define the coverage obtained by omitting that test:

$$\widetilde{V}(S) = \sum_{j=1}^{m} \max_{d_i \in S} b_{ji}. \tag{13}$$

This relaxation can count a requirement whose selected supporting passages all fail the matching conditions. Let

$$D_{\mathrm{app}}(S) = \sum_{j=1}^{m} \mathbf{1}[\exists d_i \in S : b_{ji} = 1]\, \mathbf{1}[\nexists d_i \in S : b_{ji} c_{ji} = 1] \tag{14}$$

count such requirements. For a fixed nonnegative rational λ, use $J_\lambda(S) = V(S) - \lambda H(S)$ to analyze a scalar preference over the two objectives. The parameter λ specifies this theoretical preference; the implemented selection procedure uses the rules in Section 3.4.

Lemma 1 (Applicability correction)**.** *Every feasible context satisfies*

$$\begin{aligned} V(S) &= \widetilde{V}(S) - D_{\mathrm{app}}(S), \\ J_\lambda(S) &= \widetilde{V}(S) - \underbrace{\left[D_{\mathrm{app}}(S) + \lambda H(S)\right]}_{\text{applicability-related correction } P_\lambda(S)}, \\ P_\lambda(S) &= D_{\mathrm{app}}(S) + \lambda H(S) \geq 0. \end{aligned} \tag{15}$$

Proof. Since $b_{ji}c_{ji} \le b_{ji}$, a requirement has one of three coverage states: neither expression covers it, both cover it, or the relaxed expression covers it while the valid-support expression does not. The last state is exactly its contribution to D_{app}. Summing over requirements gives the first identity; substituting the definition of J_λ gives the second. Equivalently,

$$D_{\mathrm{app}}(S) = \sum_{j=1}^{m} \left[\prod_{d_i \in S} (1 - b_{ji}c_{ji}) - \prod_{d_i \in S} (1 - b_{ji}) \right]. \tag{16}$$

The products are Boolean identities and require no independence assumption. Empty products equal one, so the identities also hold for the empty context. □

The correction separates two consequences of condition mismatch. D_{app} counts lost requirement coverage: adding a valid supporting passage removes that requirement's loss even if a mismatched passage remains. H separately measures IA document exposure. The indicator ι_i remains an established IA judgment; it is not defined as the complement of passage applicability.

To distinguish a relaxed upper bound from an attainable optimum, let $\overline{J} = \max_{S \in \mathcal{S}_{B,T}} \widetilde{V}(S)$ and $J^* = \max_{S \in \mathcal{S}_{B,T}} J_\lambda(S)$. Then

$$\begin{aligned} J^* &\le \overline{J}, \\ J^* = \overline{J} &\iff \exists S \in \arg\max_{S' \in \mathcal{S}_{B,T}} \widetilde{V}(S') : P_\lambda(S) = 0. \end{aligned} \tag{17}$$

Indeed, equality requires both maximum relaxed coverage and zero correction, and such a context attains the bound. Thus the relaxed bound can be unattainable even with complete information.

For any relaxed maximizer S_{rel} and true maximizer S_{opt}, substitution into Equation 15 gives

$$\begin{aligned} J_\lambda(S_{\mathrm{opt}}) - J_\lambda(S_{\mathrm{rel}}) = P_\lambda(S_{\mathrm{rel}}) - P_\lambda(S_{\mathrm{opt}}) \\ - \left[\widetilde{V}(S_{\mathrm{rel}}) - \widetilde{V}(S_{\mathrm{opt}})\right]. \end{aligned} \tag{18}$$

The relaxed choice is strictly suboptimal when its additional correction exceeds its relaxed-coverage advantage. In particular, equal relaxed coverage can conceal different valid coverage or IA exposure.

B.3 The Decision Value of Applicability Information

The preceding identity concerns a fixed context. To characterize what a selection procedure can achieve, we distinguish its retained observations from the semantic state that determines the objective. This is a finite decision problem within the classical comparison-of-experiments framework (Blackwell, 1953).

Consider a finite distribution of instances represented by (Z, Ξ). Within each observation class $Z = z$, candidate indices, requirements, the support table b, costs, and budgets are fixed and known. The state Ξ supplies the matching table c and IA indicators ι. The same finite feasible family $\mathcal{S}(z)$ is available in every state in that class. A policy restricted to Z satisfies

$$\Pr_\pi(S \mid Z, \Xi) = \Pr_\pi(S \mid Z). \tag{19}$$

It may use arbitrary computation, set selection, and independent randomization. The restriction concerns which distinctions its observations preserve. Any condition information extracted from raw text, representations, or implicit reasoning must be included in Z. For a complete representation that already determines the optimal decision, the information gap below is zero.

Define the best values with and without the additional state information:

$$\begin{aligned} J^*(z, \xi) &= \max_{S \in \mathcal{S}(z)} J_\lambda(S; \xi), \\ J_{\mathrm{full}}(z) &= \mathbb{E}[J^*(z, \Xi) \mid Z = z], \\ J_Z(z) &= \max_{S \in \mathcal{S}(z)} \mathbb{E}[J_\lambda(S; \Xi) \mid Z = z], \\ J_\pi(z) &= \mathbb{E}[J_\lambda(S_\pi; \Xi) \mid Z = z]. \end{aligned} \tag{20}$$

The full-information value permits a different context for each semantic state. The observation-based value chooses before distinguishing those states, even with their conditional distribution known exactly.

Theorem 1 (Selection loss from missing applicability information)**.** *For every positive-probability observation z and every policy satisfying Equation 19,*

$$J_{\text{full}}(z) - J_\pi(z) = \underbrace{J_{\text{full}}(z) - J_Z(z)}_{\substack{\Delta_{\text{app}}(z)\text{: missing applicability} \\ \text{information loss}}} + \underbrace{J_Z(z) - J_\pi(z)}_{\substack{R_{\text{decision}}(\pi;z)\text{:} \\ \text{selection loss given } Z}}, \tag{21}$$
$$\Delta_{\text{app}}(z) \geq 0, \qquad R_{\text{decision}}(\pi; z) \geq 0.$$

Let $W_z = \{\xi : \Pr(\Xi = \xi \mid Z = z) > 0\}$. The information term is strictly positive precisely when there is no common optimal context:

$$\Delta_{\text{app}}(z) > 0 \iff \bigcap_{\xi \in W_z} \arg\max_{S \in \mathcal{S}(z)} J_\lambda(S; \xi) = \emptyset. \tag{22}$$

The same decomposition holds after taking expectations over Z.

Proof. For any fixed context S, the statewise maximum is at least $J_\lambda(S; \xi)$. Taking conditional expectations and maximizing over the fixed choice yields $J_{\text{full}}(z) \geq J_Z(z)$. By Equation 19, $J_\pi(z)$ is a convex combination of the contexts' conditional expected values and cannot exceed $J_Z(z)$. Adding and subtracting $J_Z(z)$ gives Equation 21.

If a context is optimal in every state in W_z, choosing it attains $J_{\text{full}}(z)$ without distinguishing the states, so the gap is zero. Conversely, take a context S_0 attaining $J_Z(z)$. A zero gap implies

$$\mathbb{E}[J^*(z, \Xi) - J_\lambda(S_0; \Xi) \mid Z = z] = 0. \tag{23}$$

Every summand is nonnegative. Since the state space is finite, it must vanish in every positive-probability state. Thus S_0 is a common optimal context. Taking expectations over Z proves the final statement. □

The first term in Equation 21 measures the decision value of the missing applicability state; the second measures the policy's remaining selection loss given its observations. Perfect optimization using the same observations can remove the second term while leaving the first positive. A difference in applicability labels alone is insufficient for a strict gap: those differences must preclude a common optimal choice. This is why additional applicability information can have zero value in some selection problems.

B.4 A Tight Bound under Condition Ambiguity

We now express this information loss in terms of requirements, candidate ambiguity, and the document budget. The construction isolates uncertainty about valid uses while keeping the support structure and costs fixed.

Theorem 2 (Budgeted selection under condition ambiguity)**.** *Let m, k, B be integers with $m \geq 1$, $k \geq 2$, and $0 \leq B \leq mk$. There are k candidates $d_{j,1}, \ldots, d_{j,k}$ for each requirement r_j, with $b_{h,(j,t)} = \mathbf{1}[h = j]$. All token costs equal one and the token constraint is nonbinding. A state $\xi \in \{1, \ldots, k\}^m$ specifies the unique matching candidate d_{j,ξ_j} in each group; every other candidate has an established mismatch for that use. Suppose the retained observation is identical across these states. Then*

$$\sup_{\pi \in \Pi_Z} \inf_{\xi} \mathbb{E}_\pi[V(S; \xi)] = B/k, \qquad \max_{S \in \mathcal{S}_{B,T}} V(S; \xi) = \min(B, m), \tag{24}$$

where Π_Z contains all policies restricted to the retained observation. The optimal worst-case expected loss relative to full information is $\min(B, m) - B/k$.

Proof. For an upper bound, draw each ξ_j independently and uniformly from $\{1, \ldots, k\}$. For a fixed context S, let t_j be the number of selected documents in group j. The probability of including its unique matching document is t_j/k. Therefore

$$\mathbb{E}_\xi[V(S; \xi)] = \sum_{j=1}^{m} t_j/k = |S|/k \leq B/k. \tag{25}$$

Averaging over a policy's independent randomization preserves this bound. Hence, for every restricted policy, some state has expected coverage at most B/k.

To attain the bound in every state, select uniformly among all subsets of size B. Each of the m matching documents is included with probability $B/(mk)$, and each covers a distinct requirement. The expected coverage is consequently $mB/(mk) = B/k$. A full-information policy instead selects matching documents from $\min(B, m)$ groups. Each document covers at most one requirement, so this is optimal. □

At $B = m$, the best worst-case expected coverage is a fraction $1/k$ of the full-information optimum. For $m = B = 1$ and $k = 2$, the two states make either the first or the second document valid. A policy choosing the first with probability p obtains expected coverage p in one state and $1 - p$ in the other. The best worst-case value is $1/2$, attained by $p = 1/2$, whereas full information gives value one in both. The bound concerns the worst state for each policy; a restricted policy can still choose correctly on a particular instance.

Corollary 1 (Including IA exposure)**.** *In the construction of Theorem 2, give matching documents IA indicator zero and the remaining documents indicator one. For $\lambda \geq 0$ and an allowed empty context,*

$$\sup_{\pi \in \Pi_Z} \inf_{\xi} \mathbb{E}_\pi[J_\lambda(S; \xi)] = \max\{0,\ B/k - \lambda(1 - 1/k)\},$$
$$J^*_\lambda(\xi) = \min(B, m). \tag{26}$$

Proof. Under the uniform state distribution used above, any nonempty fixed context has expected IA fraction $1 - 1/k$. Its expected utility is $|S|/k - \lambda(1 - 1/k)$, maximized among nonempty contexts at size B when $B > 0$. The empty context gives zero, yielding the upper bound. Uniform sampling of size-B subsets gives the stated nonempty value in every fixed state, by the inclusion probabilities in the previous proof. Choose this policy or the empty context according to which has higher value. For $B = 0$ both sides are zero. The full-information optimum selects $\min(B, m)$ matching documents with zero IA exposure. □

Corollary 2 (Partial information in the two-candidate case)**.** *In the $m = B = 1$, $k = 2$ example, let the two states have equal prior probability, and let their finite observation distributions be P_1 and P_2. Then the best expected coverage using the observation is*

$$\max_{\pi(Z)} \mathbb{E}[V(S_\pi; \Xi)] = \frac{1 + \mathrm{TV}(P_1, P_2)}{2},$$
$$\mathrm{TV}(P_1, P_2) = \frac{1}{2} \sum_z |P_1(z) - P_2(z)|. \tag{27}$$

The full-information gap is $(1 - \mathrm{TV}(P_1, P_2))/2$.

Proof. Since coverage is nonnegative, replacing an empty choice with either document cannot lower its value. Let $a(z) \in [0, 1]$ be the probability of choosing the first document after observing z. Its expected coverage is

$$\frac{1}{2} \sum_z \{P_1(z)a(z) + P_2(z)[1 - a(z)]\} = \frac{1}{2} \left[1 + \sum_z (P_1(z) - P_2(z))a(z)\right]. \tag{28}$$

Set $a(z) = 1$ where $P_1(z) > P_2(z)$ and zero where the inequality is reversed. The sum of positive differences is the total variation distance, giving the maximum. Full information attains coverage one. □

Thus the information loss need not rely on exactly identical observations. Overlapping observation distributions can also leave a positive gap. The distance here measures distinguishability of semantic states, rather than Euclidean distance between document embeddings.

B.5 Complexity of Exact Selection

The information bounds concern what a policy can decide from its observations. We next give the applicability table in full and consider the computational cost of choosing a complementary context.

Theorem 3 (Selection complexity)**.** *The decision problem in Definition 2 is NP-complete when B is part of the input. This holds even when applicability is known exactly, all documents have equal token cost, no document has an IA judgment, and the token constraint is nonbinding.*

Proof. A certificate is a list of at most n distinct document indices. Given this list, the document and token budgets, covered requirements, and IA fraction can be checked by scanning the supplied tables and performing arithmetic on the encoded inputs. Verification takes polynomial time, so the problem is in NP.

For NP-hardness, we reduce from Set Cover (Karp, 1972). An instance consists of a nonempty universe $\mathcal{U} = \{u_1, \ldots, u_m\}$, subsets $\mathcal{D}_1, \ldots, \mathcal{D}_n \subseteq \mathcal{U}$, and an integer $0 \le k \le n$. The question is whether at most k subsets cover $\mathcal{U}$. Create one requirement r_j for each element u_j and one document d_i for each subset $\mathcal{D}_i$. Set

$$\alpha_{ji} = \mathbf{1}[u_j \in \mathcal{D}_i], \qquad \iota_i = 0, \quad \ell_i = 1, \quad B = k, \quad T = n, \quad c = m, \quad \varepsilon = 0. \tag{29}$$

All declared support is condition-matched; there are no IA documents. The construction writes an $m \times n$ binary table and therefore has polynomial size and takes polynomial time.

If the Set Cover instance has a cover indexed by I, select the corresponding documents in any order. Since $|I| \le k$, both budgets are satisfied. Every requirement has an applicable supporting document, so $V(q, S) = m$, while $H(q, S) = 0$.

Conversely, suppose the constructed selection instance has a feasible context satisfying Equation 11. Its coverage must equal m, since there are only m requirements. For every j, some selected d_i therefore has $\alpha_{ji} = 1$, which means $u_j \in \mathcal{D}_i$. The corresponding subsets cover $\mathcal{U}$ and number at most $B = k$. Thus the Set Cover instance is a yes-instance if and only if the selection instance is a yes-instance. The reduction establishes NP-hardness; membership in NP completes the proof. □

Corollary 3 (Exact optimization)**.** *Computing a Pareto-optimal context for Equation 2 is NP-hard. For any fixed positive rational λ, exactly maximizing $V(q, S) - \lambda H(q, S)$ over $\mathcal{S}_{B,T}(q)$ is also NP-hard.*

Proof. In the constructed instances, $H(q, S) = 0$ for every feasible context. Any context with less than maximum coverage is dominated by one with maximum coverage. Hence a Pareto-optimal context must maximize V, as must an optimizer of $V - \lambda H$. Either optimizer would decide Set Cover by checking whether its coverage equals m. □

B.6 Selection within a Finite Context Family

AppRAG constructs a finite family rather than enumerating all feasible contexts. For any nonempty $\mathcal{F} \subseteq \mathcal{S}_{B,T}$ and returned context $S_\pi \in \mathcal{F}$, write $J^*_{\mathcal{F}} = \max_{S \in \mathcal{F}} J_\lambda(S)$. Its loss separates as

$$J^* - J_\lambda(S_\pi) = \underbrace{J^* - J^*_{\mathcal{F}}}_{\text{context-family loss}} + \underbrace{J^*_{\mathcal{F}} - J_\lambda(S_\pi)}_{\text{within-family selection loss}}. \tag{30}$$

This identity separates the availability of a useful context from the ability to select it. A conditional estimation bound makes the second term explicit.

Lemma 2 (Finite-family estimation bound)**.** *Suppose a scalar estimate $\widehat{J}$ satisfies $\max_{S \in \mathcal{F}} |\widehat{J}(S) - J_\lambda(S)| \le \epsilon_{\text{est}}$, and the selected context satisfies $\max_{S \in \mathcal{F}} \widehat{J}(S) - \widehat{J}(S_\pi) \le \epsilon_{\text{opt}}$. Then*

$$J^* - J_\lambda(S_\pi) \le J^* - J^*_{\mathcal{F}} + 2\epsilon_{\text{est}} + \epsilon_{\text{opt}}. \tag{31}$$

Proof. Take $S^*_{\mathcal{F}} \in \arg\max_{S \in \mathcal{F}} J_\lambda(S)$. Adding and subtracting the two estimates gives

$$\begin{aligned} J^*_{\mathcal{F}} - J_\lambda(S_\pi) = {} & J_\lambda(S^*_{\mathcal{F}}) - \widehat{J}(S^*_{\mathcal{F}}) \\ & + \widehat{J}(S^*_{\mathcal{F}}) - \widehat{J}(S_\pi) \\ & + \widehat{J}(S_\pi) - J_\lambda(S_\pi) \le 2\epsilon_{\text{est}} + \epsilon_{\text{opt}}. \end{aligned} \tag{32}$$

Substitute into Equation 30. □

B.7 Implications for AppRAG

The analysis identifies two distinct sources of selection loss. A representation that discards decision-relevant condition information can leave a positive Δ_{app} even after its selection policy is optimized. With applicability known, several usable passages may still repeat one requirement while another supplies a missing link. This separates the role of applicability assessment from the role of budgeted context construction.

AppRAG estimates applicability, uses its predictions to prioritize candidates and verify exchanges, and constructs coverage-oriented alternatives. These operations determine the context family available for selection. It then compares drafts to assess how the fixed generator uses those contexts. The information condition concerns what a procedure can distinguish, whether through explicit labels or implicit reasoning; it does not require a particular scorer or projector architecture.

All bounds above concern the coverage–exposure objective and the stated observation models. The deployed comparison judge and risk gate have no established uniform $\widehat{J}$ error or optimization tolerance of the form assumed in Lemma 2. Likewise, full coverage need not produce a correct answer, and a generator may answer from parametric knowledge without complete retrieved support. The results therefore characterize the evidence-selection problem rather than an answer-accuracy or approximation guarantee for AppRAG.

For fixed B, the explicit-table problem permits $O(n^B)$ enumeration with polynomial work per context. The complexity result concerns variable budgets, whereas the information gap can occur even at $B = 1$. Empirical comparisons in Appendix C.10 evaluate the incremental value of the implemented signals and selection procedure; the theoretical state distributions do not specify the size of those empirical gains.

C Additional Analysis

C.1 Applicability Scores and Evidence Selection

We examine whether applicability should be used as a document-ranking score alone or as an input to subsequent evidence selection. This analysis uses a development panel of HotpotQA, MuSiQue, and 2Wiki. All methods use the same materialized candidate union, Qwen3-8B generator, and document cap $B = 4$. The full procedure may retain fewer than four documents. Scores are retrospective development results, not a new confirmation test. Support recall measures the fraction of annotated support documents selected per question; it does not measure applicability-label accuracy. Answer scoring follows Appendix **??**.

For direct ranking, we compare question-only and requirement-aware BGE-reranker-v2-m3 (Chen et al., 2024; Liu et al., 2024) with the ensemble's joint $P(\mathrm{A})$ and conditional $P(\mathrm{A})/[P(\mathrm{A})+P(\mathrm{IA})]$ scores. A constrained variant selects the lexicographically earliest set in conditional-score order matching or exceeding requirement-aware BGE's summed relevance, $P(\mathrm{A})+P(\mathrm{IA})$. Single-answer controls use the full-union candidate head and raw or calibrated query ranking, with or without verification. The ordinary-objective control retains drafts, upstream applicability signals, and the risk veto. Joint applicability improves support recall and F1 over conditional ranking; constraining relevance recovers much of the gap (Table 6), since conditional scores can mask low relevance. The head-specific F1 gain over BGE remains unresolved. Calibration improves F1 by 2.52 points with verification fixed (simultaneous 95% interval: $[0.37, 5.36]$, adjusted across six primary comparisons), while ranking-only F1 barely changes, supporting calibration with verified replacement.

The full procedure attains the highest F1 despite lower support recall than joint applicability ranking, using additional drafts and online checks. This is a system comparison, not a matched-cost gain. Changing only its final objective leaves every question's EM/F1 unchanged while retaining upstream applicability signals; the control does not isolate an extra answer gain from that objective.

Table 6: Evidence-selection policies at $B = 4$. Drafts and checks are mean generation and online verification/selection calls per question, excluding upstream retrieval and scoring.

Selection policy	F1 (%)↑	Support recall (%)↑	Drafts	Checks
BGE (2024), question only	31.12	73.98	1.00	0
BGE (2024), question + requirements	27.56	69.78	1.00	0
Joint applicability $P(\mathrm{A})$	32.25	**77.72**	1.00	0
Conditional applicability	20.88	50.15	1.00	0
Conditional + relevance constraint	28.41	72.47	1.00	0
AppRAG candidate head	31.55	76.54	1.00	0
Raw query vector	25.21	65.34	1.00	0
Calibrated query vector	25.12	68.64	1.00	0
Raw vector + verifier	30.21	69.35	1.00	2
Calibrated vector + verifier	32.73	72.44	1.00	2
AppRAG (full procedure)	**38.55**	**73.46**	**2.94**	**4**
Full procedure, ordinary objective	38.55	73.18	2.94	4

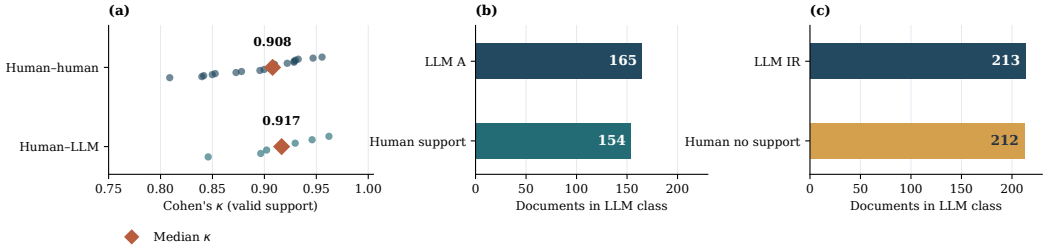


Figure 8: Human comparison of automatic A/IR labels. (a) Pairwise human agreement on valid support across 637 passages (21 pairs) and individual human–LLM agreement on the 378 A/IR passages (seven annotators); diamonds mark medians, with their values printed above. (b) All 165 automatic A passages versus the 154 with human-majority valid support (10 without support; one unresolved). (c) All 213 automatic IR passages versus the 212 with human-majority no support (one with support). IA and U are outside this binary comparison.

C.2 Human Evaluation

Seven annotators assessed valid support for 637 passages from 120 questions, with LLM labels revealed after each row was filled; we compare their 4-of-7 majority with the saved A/IR judgments on 378 passages. Human support judgments were consistent (median pairwise $\kappa = 0.908$), while automatic A and IR aligned with its positive and negative sides, respectively (154/165 and 212/213 passages). This supports the A/IR distinction in these saved context judgments; it does not establish separate downstream gains for either label or validate the distinct training-teacher annotations.

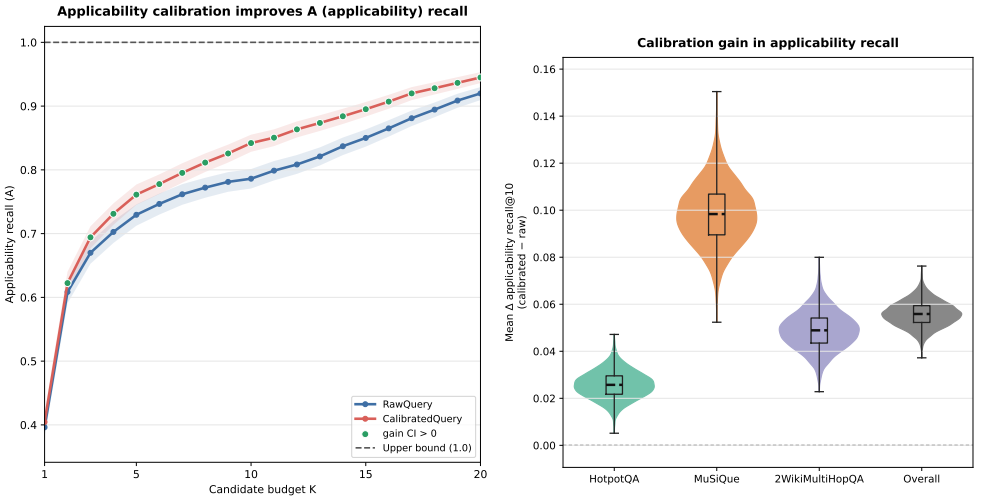
C.3 Comparison of Applicability Training Objectives

We compare broad relevance, pointwise applicability, within-question A-over-IA ranking, and an equal mixture of the latter two using Qwen3-8B answer F1 on the development panel in Appendix C.1. All use Qwen3-Reranker-4B with fresh rank-8 LoRA adapters, identical question–requirements–document inputs and prediction heads, and shared relevance and support auxiliaries. Each uses three matched initializations and one epoch over 1,799 training questions, with no question-ID or exact normalized-question overlap with evaluation. Applicability targets combine historical teacher labels with annotated training supports as positives. New Qwen3.6-35B-A3B annotations supply broad relevance targets only; their A/IA labels are excluded because they mislabel valid bridge evidence as IA. These weak-supervision controls share an architecture rather than exactly retraining the deployed ensemble and candidate head. Ensembles average three ranking scores over the same union, with fixed loss weights and training budgets and no objective-specific hyperparameter search.

The mixed objective leads at $B = 2$, while broad relevance leads at $B = 4$. At $B = 4$, all adjusted difference intervals include zero, resolving neither superiority nor equivalence (Table 7). Intervals use 10,000 question-paired bootstrap resamples stratified by dataset and sampling seed, conditional on fitted models; individual ranges capture initialization variation. A-over-IA ranking varies most, with only 109 training questions containing both labels. These controls favor retaining relevance and assessing answer quality, but do not isolate a mixed-loss advantage.

Table 7: Training objectives on the shared development panel (F1, %). Ranges cover three individual initializations. Mixed-objective differences use ensemble $B = 4$ results with Bonferroni-adjusted simultaneous 95% intervals for three comparisons.

Training objective	**Ensemble F1↑**		**Individual F1 range**
	$B = 2$	$B = 4$	$B = 4$
Broad relevance	32.37	**33.44**	30.80–33.03
Pointwise applicability	33.96	32.52	31.34–33.09
A-over-IA ranking	33.74	31.05	26.05–32.44
Pointwise + A-over-IA (mixed)	**34.11**	**33.26**	**30.12–32.06**
Comparison	**ΔF1**	**Adjusted 95% interval**	
Mixed − relevance	−0.19	[−3.43, 3.03]	
Mixed − pointwise	+0.74	[−1.67, 3.10]	
Mixed − A-over-IA	+2.21	[−1.33, 5.81]	



(a) Applicability Recall@K across ranking depths. (b) Mean paired applicability Recall@10 gains.

Figure 9: Query calibration with fixed candidates and document embeddings. (a) Shading shows pointwise 95% bootstrap intervals; green markers indicate positive paired gain intervals. (b) Violins show bootstrap distributions of the mean gain, with quartile boxes, min–max whiskers, and mean lines. Here K is ranking depth within the fixed pool, not the final context budget.

C.4 Applicability Recall under Query Calibration

We test whether query calibration brings applicable evidence forward while holding candidates and document embeddings fixed. The study uses 1,800 questions from HotpotQA, MuSiQue, and 2Wiki, with three sampling seeds and 600 questions per dataset. A frozen rank-64 projector reranks the same BGE-M3/Qwen3-Embedding-4B union using fixed BGE-M3 document vectors. Pools contain 14–45 candidates; normalized inner products determine rank, with document-ID tie-breaking. Reference-aware Qwen3-32B dual-view teacher labels are joined after ranking and are not replay inputs or human gold labels.

Applicability Recall@K is the fraction of the pool's A-labeled documents ranked in the top K. Excluding queries without A candidates leaves 596/525/599 questions from HotpotQA/MuSiQue/2Wiki; Overall weights all 1,720 equally. Evaluation precedes verification and generation.

Calibration improves Recall@10 by 5.59 points (paired 95% interval: 4.59–6.63), with positive paired intervals on all three datasets. MuSiQue has the lowest initial recall and largest gain. Benefits diminish as deeper rankings approach full recall; pointwise gain intervals are positive at $K = 2$–20 but include zero at $K = 1$. The violins describe uncertainty in mean gains, not gains for every query. Thus calibration brings applicable evidence forward within a fixed pool; Appendix C.1 assesses its downstream benefit when combined with verified exchange.

C.5 Compatibility with Embedding Models

Retrieval quality. Table 8 reports support-document Recall@B and nDCG@B gains with two embedding models. Both improve on average, especially at $B = 2$, where recall rises by 7.52 points with BGE-M3 and 5.37 with Qwen3-Embedding-4B, versus 1.97 and 0.93 at $B = 4$. The nDCG gains follow the same pattern, indicating better coverage and ordering under the smaller context budget. At $B = 4$, dataset-level gains and losses coexist, as detailed below.

Table 8: AppRAG retrieval gains (pp).

B	ΔRecall@B	ΔnDCG@B
BGE-M3 (2024)		
2	+7.52	+6.88
4	+1.97	+3.59
Qwen3-Embedding-4B (2025b)		
2	+5.37	+4.40
4	+0.93	+1.89

To assess compatibility across embedding models, we compare AppRAG with Vanilla RAG under BGE-M3 and Qwen3-Embedding-4B, holding the embedding, question, and generator fixed within each pair. This evaluation uses 600 questions per dataset on HotpotQA, MuSiQue, and 2Wiki. The six generators are Qwen3-0.6B, 1.7B, 4B, 8B, 14B, and LLaMA-3.1-8B (Grattafiori et al., 2024). Table 8 averages support-document Recall@B and nDCG@B gains equally over datasets, generators, and candidate ranking depths $K \in \{5, 7\}$ for each per-context document budget B. Recall measures the fraction of annotated support documents in the selected context; nDCG uses binary support labels and logarithmic rank weights. Table 9 additionally averages over $B \in \{2, 4\}$ for the embedding rows.

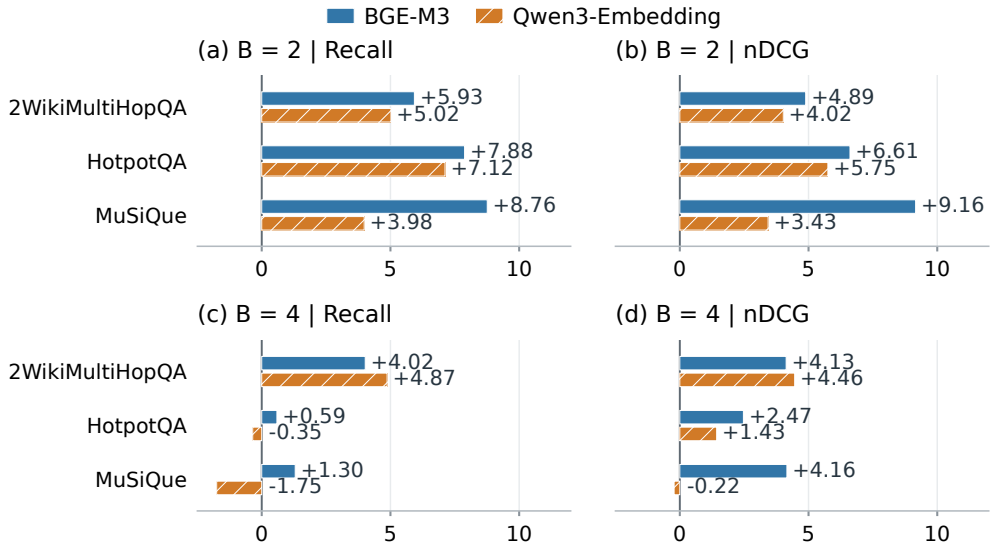


Figure 10: AppRAG gains in support-document Recall@B and nDCG@B (pp), averaged over generators and candidate ranking depths for each embedding model.

Within this development sample, both retrieval metrics improve on all three datasets at $B = 2$. At $B = 4$, Qwen3-Embedding-4B loses recall on HotpotQA and MuSiQue (-0.35 and -1.75

points); MuSiQue nDCG falls by 0.22 points. These averages characterize the sampled development questions. For answer quality, all 36 embedding–generator–dataset F1 gains are positive; 35 EM gains are positive and one is zero.

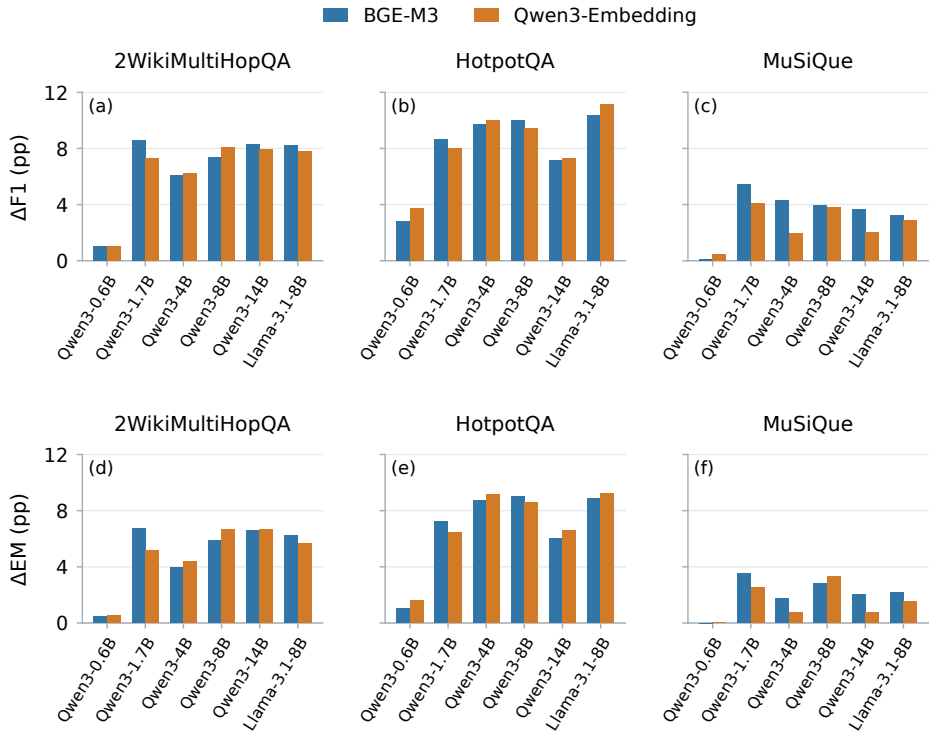


Figure 11: AppRAG F1 and EM gains (pp) by embedding model and generator, averaged over candidate ranking depths and document budgets within each dataset.

C.6 Compatibility with Rerankers

Answer quality. Table 9 shows mean F1 gains of 6.05 and 5.73 points with the two embeddings, alongside positive EM gains. On a separate development panel, AppRAG improves each reranker's mean F1 by 7.72–9.29 points and EM by 5.37–7.04, supporting compatibility across retrieval components. The two groups use different samples, so their gain magnitudes are not directly comparable.

Table 9: AppRAG answer-quality gains (pp).

Model	ΔF1	ΔEM
Embedding		
BGE-M3 (2024)	+6.05	+4.61
Qwen3-Embedding-4B (2025b)	+5.73	+4.42
Reranker		
BGE-reranker-v2-m3 (2024)	+7.72	+5.37
GTE-multilingual (2024)	+7.95	+6.11
Qwen3-Reranker-0.6B (2025b)	+9.29	+7.04

We add AppRAG to each reranking pipeline and compare against that reranker alone on HotpotQA, MuSiQue, and 2Wiki. The rerankers are BGE-reranker-v2-m3 (Chen et al., 2024; Liu et al., 2024), GTE-multilingual-reranker-base (abbreviated as GTE-multilingual in the table), and Qwen3-Reranker-0.6B. This study uses 30 questions per dataset from the development panel and the same six generators as the embedding study. Table 9 averages F1 and EM differences equally over datasets and generators; the figure averages over datasets for each generator separately. These questions had been used during development, so results measure compatibility within this panel. No significance tests accompany the summary means. With Qwen3-0.6B as generator, EM is unchanged and F1 changes are near zero, with both signs across rerankers. For each larger generator, all three rerankers have positive mean F1 and EM gains across datasets.

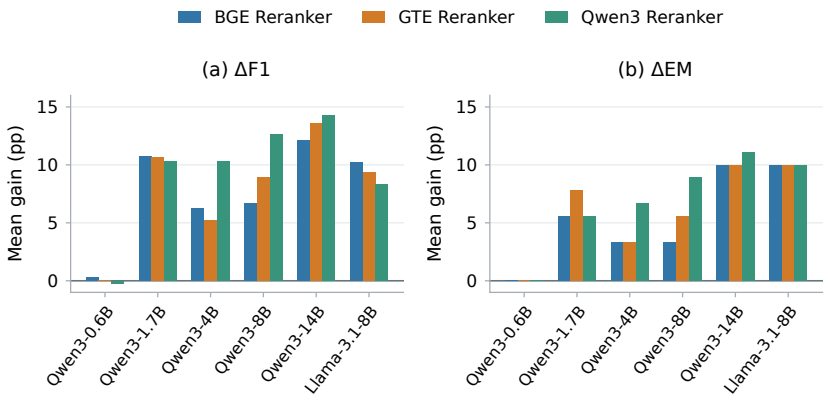


Figure 12: AppRAG F1 and EM gains (pp) by reranker and generator, averaged equally across datasets.

C.7 Sensitivity to Candidate Ranking Depth and Context Budget

Setup. We vary ranking depth $K \in \{5, 7, 10\}$ and context budget $B \in \{2, 4\}$ with the six generators in Appendix C.5 and both embedding models. All configurations use the same 1,765 questions (600 HotpotQA, 600 MuSiQue, 565 2Wiki), excluding 35 2Wiki questions with fewer than ten candidates. Scores average over embeddings within each question, then over pooled questions. This sample is separate from the main-table evaluation.

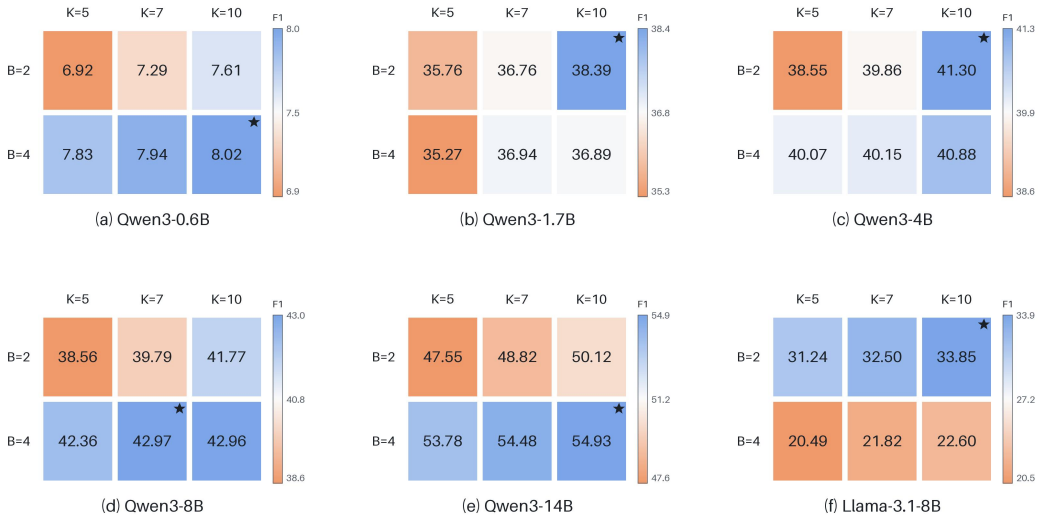


Figure 13: AppRAG F1 (%) across candidate ranking depths K and per-context document budgets B. Stars mark the highest observed score for each generator; color scales are independent across panels.

Figure 13 shows that deeper candidate rankings help most under tight context budgets: increasing K from 7 to 10 gives every generator a larger F1 gain at $B = 2$ than at $B = 4$. For Qwen3-8B, the budget-averaged gain is 0.99 points. More candidates thus offer useful alternatives without requiring a larger answer context. The preferred budget varies by generator: at $K = 10$, Qwen3-14B favors $B = 4$, whereas LLaMA-3.1-8B, Qwen3-1.7B, and Qwen3-4B peak at $B = 2$. Stars denote grid-specific maxima.

Table 10: Answer refinement and AppRAG across retrieval pipelines (%).

RAG system	Method	EM↑	F1↑	Valid judgments
Vanilla RAG (2020)	CoVe-2Step (2024)	23.70	29.08	93.70
	Self-Refine-1Iter (2023)	15.93	21.79	94.44
	AppRAG	**30.37**	**35.55**	98.15
EntityLinkRAG (2026)	CoVe-2Step (2024)	15.19	19.51	93.70
	Self-Refine-1Iter (2023)	11.48	15.72	96.67
	AppRAG	**28.89**	**33.05**	97.04
RRF-ensemble (2009)	CoVe-2Step (2024)	25.93	32.52	94.07
	Self-Refine-1Iter (2023)	19.26	26.39	92.22
	AppRAG	**30.37**	**35.55**	98.15

Bold marks the best EM/F1 per pipeline. Valid judgments measure evaluator completion, not answer quality.

C.8 Comparison with Verifier-based RAG

We compare evidence selection and answer refinement on 90 questions each from HotpotQA, MuSiQue, and 2Wiki, using three interfaces in Table 10. Native and AppRAG answers use Qwen3-8B; evidence-grounded CoVe (Dhuliawala et al., 2024) and Self-Refine (Madaan et al., 2023) revise native answers with Qwen3-32B under deterministic decoding with thinking disabled. CoVe uses three calls to plan checks, answer them without the draft, and revise; Self-Refine uses one two-call feedback–refinement iteration. Native contexts contain four documents; reused AppRAG outputs use two or four. Refiners abstain on insufficient support; all outputs count in EM/F1.

EM/F1 scoring lowercases answers, deletes ASCII punctuation, removes English articles, and maximizes over reference aliases. Punctuation deletion rather than space replacement (Appendix **??**) explains small F1 differences from the main table. Qwen3.6-35B-A3B semantic scores are omitted because valid judgments fall below the 98% coverage threshold.

Answer quality. AppRAG has the highest observed EM and F1 in every retrieval pipeline (Table 10), exceeding CoVe by 3.03–13.54 F1 points. This compares complete procedures: AppRAG changes the evidence context, while refiners retain native contexts and frequently abstain; AppRAG produces no empty answers. The results therefore reflect evidence selection versus refinement together with their different abstention behavior.

Table 11: Closely related adaptations with Qwen3-8B (%). Bold marks the best score per column.

Method	HotpotQA		MuSiQue		2Wiki		Overall	
	EM	F1	EM	F1	EM	F1	EM	F1
AppRAG-frozen	**55.56**	**63.96**	17.78	24.67	**17.78**	**18.84**	30.37	35.82
SetR-IRI-B4[†] (2025)	53.33	59.84	**27.78**	**34.05**	16.67	16.98	**32.59**	**36.96**
TOUR-soft-1step-full (2023)	47.78	54.34	16.67	21.30	11.11	11.43	25.19	29.02
SpecRAG-4Draft-k2 (2025)	32.22	38.09	8.89	15.56	15.56	16.43	18.89	23.36

[†] SetR uses Qwen3-32B for evidence selection.

C.9 Comparison with Closely Related Methods

We test AppRAG against requirement-based set selection, query adaptation, and multi-draft verification using adaptations of SetR (Lee et al., 2025), TOUR (Sung et al., 2023), and Speculative RAG (Wang et al., 2025). All use the same 90 questions from each of HotpotQA, MuSiQue, and 2Wiki, Qwen3-8B answer generation with greedy decoding, and at most four final passages. SetR uses a Qwen3-32B selector; TOUR performs one query-vector update followed by full-corpus retrieval; SpecRAG generates four two-passage drafts and selects with Qwen3-32B. AppRAG reuses its saved outputs. We report EM/F1 and equal-weight dataset means.

In Table 11, AppRAG exceeds TOUR and SpecRAG by 6.80 and 12.46 Overall F1 points (both Holm-adjusted $p < 0.01$), and achieves the highest EM/F1 on HotpotQA and 2Wiki. SetR leads on MuSiQue and scores 1.14 Overall F1 points above AppRAG; that paired difference is not statistically resolved. The comparison supports AppRAG's answer quality relative to query-feedback and multi-draft alternatives under these configurations. Together with the gains across eight RAG integrations (§ 4.2) and different retrieval components (§ C.5), these results support reusing AppRAG's evidence-selection procedure across retrieval interfaces.

C.10 Applicability Beyond Set Selection

To test whether applicability adds value beyond relevance and requirement-based set selection, we use three controlled comparisons. On 1,350 retrospective questions across HotpotQA, MuSiQue, and 2Wiki, a seven-arm study varies ordinary versus applicability-trained offline scores and online verification in a controlled AppRAG pipeline, and supplies either signal to a SetR-style selector. An eight-arm study on 600 fresh 2Wiki questions compares single and generic multi-draft SetR-style selection, an AppRAG-informed SetR-style hybrid, and controlled AppRAG under shared inputs. The hybrid and generic multi-draft arms have the same call limits. A separate set of 120 construction-defined mixed-use cases tests whether frozen models distinguish a passage's valid use for one requirement from its mismatched use for another.

On the retrospective set, applicability yields small, statistically unresolved F1 gains in both controlled AppRAG and SetR-style selection. On fresh 2Wiki, the applicability hybrid's gain over its ordinary counterpart is also unresolved. It exceeds controlled AppRAG by 3.00 F1 points (95% CI $[1.43, 4.66]$), using eight rather than four judge calls, while generic multi-draft SetR-style selection scores slightly higher. The retrospective hybrid gains over neither ordinary nor generic multi-draft controls. In mixed-use cases, applicability-trained support slots better rank valid above mismatched uses (+7.9 points, $[5.0, 11.2]$). The risk residual improves mixed-use versus safe-passage ranking for the raw scorer but reduces it for the calibrated head. Applicability thus provides measurable requirement-use discrimination within budgeted set selection, while answer-level gains remain setting-dependent.

We compare AppRAG-guided selection with a generic full-pool multi-draft SetR-style adaptation on the frozen 600-query 2WikiMultiHopQA cohort. Both use Qwen3.6-35B-A3B for selection, Qwen3-8B for generation, and eight judge calls per query; token counts include judge input only.

Table 12: Answer quality and judge input cost under matched selection-call budgets.

Method	F1 ↑	Judge input tokens/query ↓	Judge calls/query ↓
Generic multi-draft SetR-style	26.02	29.71k	8
AppRAG-guided SetR-style	25.77	**21.66k**	8

AppRAG's constrained candidate sets reduce judge input tokens by 27.1% at the same call count. The two methods have identical F1 on 598/600 queries; the other two favor generic selection, producing a 0.25-point mean F1 difference. AppRAG uses 1.38 versus 1.27 generator calls per query, so this result establishes judge-input savings rather than lower total inference cost.

C.11 Cost and Cost-Efficiency Analysis

We measure AppRAG's online LLM input and output tokens on 350 MuSiQue and 730 2WikiMultiHopQA questions at $B \in \{2, 4, 8\}$, using Qwen3-8B for requirements and answer drafts and Qwen3.6-35B-A3B for verification and selection. The budget comparison uses equal-dataset mean stage counts and charges the same upstream requirements and verification at each budget; embedding, reranking, and offline preparation are outside this token accounting. Increasing B from 2 to 8 raises mean usage from 11,170 to 18,109 tokens per query as drafting and selection consume more tokens (Figure 14). The verification share consequently falls from 79.8% to 49.2%, while its absolute token count remains unchanged. A smaller context budget therefore reduces downstream workload but leaves a substantial upstream verification cost.

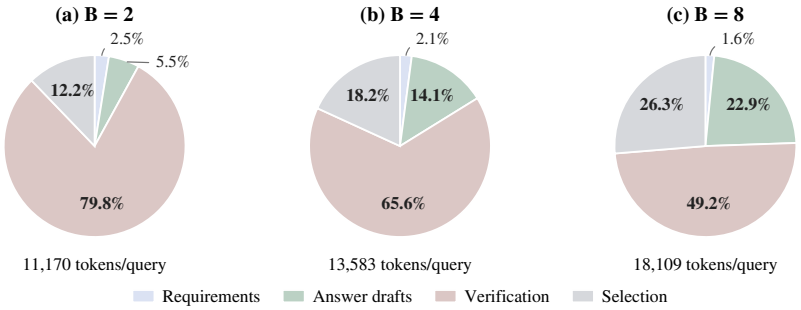


Figure 14: AppRAG's online LLM token composition across document budgets on the same 1,080 questions. Stage counts are averaged with equal weight across datasets before normalization; totals below the pies use the same averaging.

C.12 Answer-Invariant Risk Gating

We compare accepting all proposed evidence sets with an answer-invariant Risk gate on two non-overlapping subsets drawn from the same three QA benchmarks: Subset A contains 1,800 questions and Subset B contains 1,753 questions. The gate preserves macro F1 in both subsets, at 42.572 and 41.743, respectively. It reduces mean per-query IA document share from 0.764% to 0.736% in Subset A and from 1.469% to 1.241% in Subset B; this share measures the composition of selected evidence, not the absolute number of IA documents.

Table 13: Answer-invariant Risk gating on two evaluation subsets.

Subset	Strategy	Macro F1 ↑	Mean IA share ↓
A ($n = 1{,}800$)	No gate	42.572	0.764%
	Answer-invariant Risk gate	42.572	**0.736%**
B ($n = 1{,}753$)	No gate	41.743	1.469%
	Answer-invariant Risk gate	41.743	**1.241%**

C.13 Additional Case Study

We inspect saved Qwen3-8B traces for the MuSiQue embassy and HotpotQA film cases with BGE-M3 retrieval, $K = 7$, and initial budget $B = 4$. Cases were selected after inspecting outcomes and evidence, without rerunning retrieval or generation. Each final two-passage context retains one BGE support and adds a complementary passage. BGE is an independent baseline; candidate access and final context sizes differ across pipelines. Rank arrows compare initial applicability and calibrated union rankings, without isolating the projector. Colors and IA counts use text inspection, not verifier labels. Both final proposals pass the risk gate. These selected traces illustrate evidence revision rather than estimate average gains or component effects.

For the embassy case (Figure 6), the retained bridge is textually supported but lacks an official support annotation. Vanilla RAG also answers France without either audited wrong-city passage, so its error cannot be attributed to those passages.

Recovering missing support. The question asks for the actress in the Mel Damski film based on an Alabama murderer (Figure 15). BGE retrieves *Wife, Mother, Murderer* but lacks its cast link; the other passages fail the murderer condition. All three rerankers and AppRAG's initial context S_{app} yield Marie Hilley, the person portrayed, while Vanilla RAG returns Judith Light. Calibration moves the cast passage from seventh to third; verified exchange adds it to S_{cal}, correcting the answer to Judith Light. Final selection retains the film and cast passages in S^*, preserving the answer with two passages. Both comparison calls use the same alternative order, leaving order sensitivity untested in this trace.

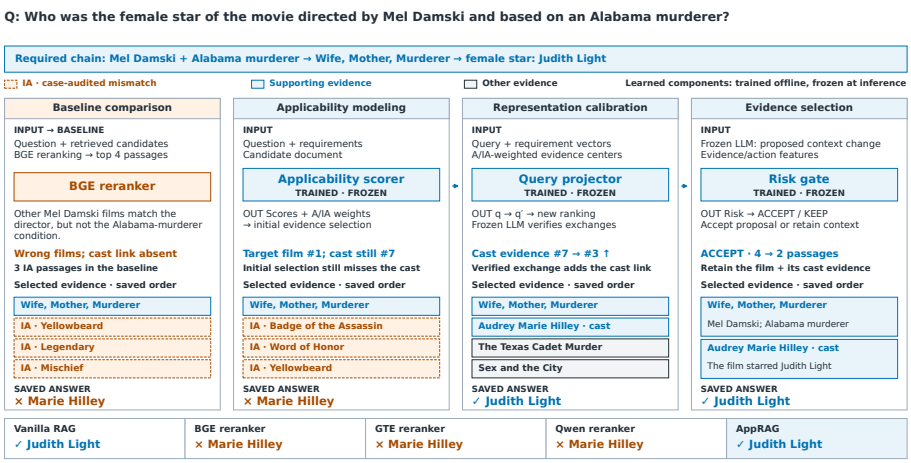


Figure 15: HotpotQA film case: AppRAG recovers missing cast evidence and retains the film and cast passages, correcting the answer to Judith Light. Colors follow Figure 6.

Table 14: Inputs and supervision for the trainable components.

Trained component	Input	Supervision
Scorer adapters and heads (θ)	Question, fixed requirements, passage title and text	Teacher A/IA/IR/U and coverage; official support
Candidate head (η)	Frozen, standardized $\mathbf{h}_d$	Relatedness, A, IA, official support; within-question A/IA pairs
Query projector (ω)	$\mathbf{z}$, original query and fixed document embeddings	Teacher A/IA/IR/U and covered requirement IDs
Action-risk head (ψ)	Standardized context/action features $\mathbf{x}$	Increase in teacher-labeled IA fraction

D Training, Inference, and Computational Complexity

D.1 Training Inputs and Objectives

Separate fitting objectives. We fit the four parameter groups $\theta, \eta, \omega, \psi$ by minimizing separate objectives with frozen upstream features:

$$\mathcal{L}_{\mathrm{sc}}(\theta) = \beta_{\mathrm{cls}}\mathcal{L}_{\mathrm{CE}} + \beta_{\mathrm{bin}}\mathcal{L}_{\mathrm{bin}} + \beta_{\mathrm{cov}}\mathcal{L}^{\mathrm{sc}}_{\mathrm{cov}}, \tag{33}$$

$$\mathcal{L}_{\mathrm{cand}}(\eta) = \tfrac{1}{2}(\mathcal{L}_{\mathrm{point}} + \mathcal{L}_{\mathrm{pair}}), \tag{34}$$

$$\mathcal{L}_{\mathrm{proj}}(\omega) = \mathcal{L}_{\mathrm{list}} + \mathcal{L}_{\mathrm{sep}} + \mathcal{L}^{\mathrm{proj}}_{\mathrm{cov}} + \lambda_{\mathrm{disp}}\mathbb{E}[1 - \mathbf{u}'^{\mathsf{T}}\mathbf{u}], \tag{35}$$

$$\mathcal{L}_{\mathrm{risk}}(\psi) = \mathcal{L}_{\mathrm{WBCE}}(\rho_\psi(\mathbf{x}), y_{\mathrm{risk}}) + \lambda_{\mathrm{risk}}\mathcal{R}_2(\psi). \tag{36}$$

Here CE and WBCE denote class-weighted cross-entropy and positive-class-weighted binary cross-entropy, respectively, and $\mathcal{R}_2$ regularizes the risk-head weights. Equation 9 groups the projector's listwise and separation losses as $\mathcal{L}_{\mathrm{rank}}$; its expanded form is Equation 35. These are separate fits rather than a joint end-to-end loss. The targets and individual terms are defined below.

Table 14 lists their inputs and supervision. Teacher annotations are training targets, not input features. Official supporting-document labels supervise the scorer and candidate head; reference answers are not used by these fitting objectives. The generator, embedding encoders, online verifier, and comparison judge remain frozen.

Applicability scorer. For each $(q, \mathcal{R}(q), d)$, the scorer predicts a class distribution, binary evidence properties, and requirement-coverage probabilities. The target class y^{cls}_d is A, IA, IR, or U. The binary targets are official support, A, IA, relatedness, and IA conditional on relatedness. Relatedness

is positive for A/IA; the conditional IA target is evaluated only on A/IA examples. The coverage vector $\mathbf{y}_d^{\mathrm{cov}}$ records the teacher's covered requirement IDs, with padded slots masked out.

In Equation 33, $\mathcal{L}_{\mathrm{CE}}$ is class-weighted cross-entropy, $\mathcal{L}_{\mathrm{bin}}$ is masked positive-class-weighted binary cross-entropy over the binary tasks, and $\mathcal{L}_{\mathrm{cov}}^{\mathrm{sc}}$ is the corresponding masked coverage loss. Each masked loss is divided by the number of valid entries. Class weights and feature statistics are computed from the fitting partition. Each ensemble member updates only its LoRA adapters and prediction heads; its backbone weights stay fixed.

Candidate head. The head receives $\mathbf{h}_d$ from the frozen assessment pipeline. Let $o_\eta(d) = [g_\eta(\mathbf{h}_d)]_{\mathrm{app}}$ be its applicability logit, so $s(d) = \sigma(o_\eta(d))$. The pointwise term in Equation 34 averages positive-class-weighted binary cross-entropy over relatedness, A, IA, and official support. For the set $\mathcal{P}$ of within-question teacher-labeled A/IA pairs, the ranking term is

$$\mathcal{L}_{\mathrm{pair}} = \frac{1}{|\mathcal{P}|} \sum_{(q,d^+,d^-)\in\mathcal{P}} \log\bigl(1 + \exp\bigl(o_\eta(d^-) - o_\eta(d^+)\bigr)\bigr)\,. \tag{37}$$

The pointwise and pairwise terms have equal weight. Ranking supervision uses logits rather than differences between sigmoid probabilities.

Query projector. For one training question, let $\mathcal{A}$, $\mathcal{I}$, and $\mathcal{O}$ denote its teacher-labeled A, IA, and IR/U candidates. Let $\mathcal{J}_d$ be the stored requirement IDs covered by d and $t_d = t(d)$ the score after projection. Questions without an A candidate are omitted from this fitting objective. Define the target distribution and listwise loss as

$$\pi_d = \frac{\mathbf{1}[d \in \mathcal{A}](1 + |\mathcal{J}_d|/m)}{\sum_{e\in\mathcal{A}}(1 + |\mathcal{J}_e|/m)}, \qquad L_{\mathrm{list}}(q) = -\sum_{d\in\mathcal{C}} \pi_d \log \frac{\exp(t_d/\vartheta)}{\sum_{e\in\mathcal{C}}\exp(t_e/\vartheta)}, \tag{38}$$

where ϑ is the training temperature. Applicability and coverage enter the target weights; document embeddings receive no gradient updates. With $\mathrm{sp}(v) = \log(1 + \exp v)$, margin δ, and $\bar{t}_{\mathcal{A}} = \log\sum_{d\in\mathcal{A}}\exp(t_d) - \log|\mathcal{A}|$, the separation term is

$$L_{\mathrm{sep}}(q) = \sum_{\substack{\mathcal{N}\in\{\mathcal{I},\mathcal{O}\}\\ \mathcal{N}\neq\emptyset}} \mathrm{sp}\left(\max_{d\in\mathcal{N}} t_d - \bar{t}_{\mathcal{A}} + \delta\right). \tag{39}$$

For each retained coverage index j, let $\mathcal{A}_j = \{d \in \mathcal{A} : j \in \mathcal{J}_d\}$. Only indices with both a covering and a non-covering candidate contribute:

$$L_{\mathrm{cov}}^{\mathrm{proj}}(q) = \frac{1}{|\mathcal{J}_{\mathrm{fit}}|} \sum_{j\in\mathcal{J}_{\mathrm{fit}}} \mathrm{sp}\left(\max_{d\notin\mathcal{A}_j} t_d - \max_{d\in\mathcal{A}_j} t_d + \delta\right). \tag{40}$$

This term is zero when $\mathcal{J}_{\mathrm{fit}}$ is empty. The three $\mathcal{L}$ terms in Equation 35 average their corresponding per-question L terms. The displacement penalty keeps the normalized updated query near its original embedding.

Action-risk head. Training records pair an incumbent S_{cal} with a candidate action S' constructed by the frozen preceding stages. Let $H_{\mathrm{teach}}(q, S)$ be the fraction of teacher-labeled IA documents in S. The binary target is

$$y_{\mathrm{risk}} = \mathbf{1}[H_{\mathrm{teach}}(q, S') > H_{\mathrm{teach}}(q, S_{\mathrm{cal}})]\,. \tag{41}$$

The head is $\rho_\psi(\mathbf{x}) = \sigma(\mathbf{w}_\psi^{\mathsf{T}}\mathbf{x} + b_\psi)$ on standardized features, and $\mathcal{R}_2(\psi) = \|\mathbf{w}_\psi\|_2^2/\dim(\mathbf{x})$; the bias is not regularized. Weighted binary cross-entropy assigns positive examples weight N_-/N_+, where N_+ and N_- count positive and negative fitting records. Inputs exclude drafts, reference answers, and answer-correctness labels.

D.2 Training and Inference Algorithms

Algorithms 1–2 summarize fitting and frozen inference. Objectives and fitting settings appear in Appendices D.1 and E.3.

Algorithm 1 Offline training of AppRAG

Input: Four fitting collections; frozen encoders and teacher; support labels
Output: Fitted $\theta, \eta, \omega, \psi$ and normalization statistics

Cache question-only requirements and teacher class/coverage targets
for each scorer fitting fold **do**
  Fit θ with $\mathcal{L}_{\mathrm{sc}}$; freeze and save held-out predictions
end for
Cache $\mathbf{h}_d$; fit normalization and construct within-question A/IA pairs
Fit η with $\mathcal{L}_{\mathrm{cand}}$; freeze θ, η
Cache $\mathbf{z}$, $\mathbf{u}$, $\{\mathbf{v}_d\}$; fit ω with $\mathcal{L}_{\mathrm{proj}}$ and freeze
Construct incumbents and candidate actions; cache $\mathbf{x}$, y_{risk}
Fit risk-feature normalization and ψ with $\mathcal{L}_{\mathrm{risk}}$
Freeze parameters, statistics, and inference settings
return $\theta, \eta, \omega, \psi$ and normalization statistics

Algorithm 2 Frozen inference with AppRAG

Input: q; retrieval interface; frozen models; depth K, document cap B, gate τ
Output: Context S^* and answer $\hat{y}$

Derive $\mathcal{R}(q)$; retrieve $\mathcal{C}_0(q) \subseteq \mathcal{C}(q)$
Compute embeddings, assessments, $\mathbf{h}_d$, and $s(d)$
$S_{\mathrm{app}} \leftarrow \mathrm{Top}_B(\mathcal{C}_0; s)$; anchor $a \leftarrow$ first document
Compute centers, $\mathbf{z}$, $\mathbf{u}'$, $t(d)$; $E_K \leftarrow \mathrm{Top}_K(\mathcal{C}; t) \cup \{a\}$
Cache verifier outputs; $S \leftarrow S_{\mathrm{app}}$
while an admissible exchange strictly increases $\boldsymbol{\kappa}(S)$ **do**
  Apply the best non-anchor replacement from E_K (Equation 7)
end while
$S_{\mathrm{cal}} \leftarrow S$; build anchor-free calibrated/fused coverage alternatives
$\mathcal{F}(q) \leftarrow \mathrm{Deduplicate}(S_{\mathrm{cal}}, \text{alternatives})$ by document order
for $S \in \mathcal{F}(q)$ **do**
  Check serialized input length; cache $\hat{y}_S \leftarrow G(q, S)$
end for
$\widetilde{S} \leftarrow S_{\mathrm{cal}}$
if $|\mathcal{F}(q)| > 1$ **then**
  Compare anonymous alternatives in both orders; set $\widetilde{S}$ on valid agreement
end if
Compute $\mathbf{x}$; obtain S^* by Equation 8
return $(S^*, \hat{y}_{S^*})$

Inference assumes a nonempty pool and uses no reference answers or training labels. Exchanges admit unselected candidates and use the tie rules in § 3.3; verifier outputs and distinct-context drafts are cached. The serialized input limit includes prompt overhead and is distinct from T, the evidence-only allowance in the theoretical formulation. Overlength inputs are rejected (Appendix E.2). Projection triggers no new search; ACE-RAG adds playbook-conditioned final generation.

D.3 Computational Complexity

Let $n = |\mathcal{C}(q)|$, $m = |\mathcal{R}(q)|$, D be embedding dimension, r_{p} projector rank, $e = |E_K| \le K + 1$, and R the number of supplied rankings. Let F_0 count contexts before deduplication. The following bounds separate cached-feature arithmetic from neural computation and corpus indexing.

Candidate processing and selection. Similarity features cost $O(nmD)$, the P_η-parameter candidate head $O(nP_\eta)$, centers and dot products $O(nD)$, projection $O(Dr_{\mathrm{p}})$, and rank fusion/sorting $O(Rn + n \log n)$. For E scorer members with per-document forward cost C_{sc}, assessment adds

$O(EnC_{\mathrm{sc}})$; other neural-model computation is separate. Each exchange pass tests at most $(B-1)e$ replacements, copying contexts and recomputing signatures in $O(B^2e)$ time. Writing each cached document contribution as $\mathbf{k}_d$ gives $\boldsymbol{\kappa}(S) = \sum_{d\in S} \mathbf{k}_d$; each accepted replacement strictly increases its slot's contribution lexicographically. Each non-anchor slot visits at most its initial document and e candidates, bounding accepted exchanges by $(B-1)e$ and total exchange time by $O(B^3e^2)$. Coverage construction costs $O(F_0BKm)$ with cached masks and rank lookups, or $O(F_0BK(m+n))$ for the implemented list-based lookups and fallbacks. These bounds concern the restricted candidate family, not global selection (Appendix B).

Offline fitting and storage. Let N_{sc} count scorer examples across members and epochs, with forward/backward cost $C_{\mathrm{sc}}^{\mathrm{fb}}$ each. The candidate head processes N_{c} examples and P_{c} pairs over U_{c} full-batch updates; the projector sees N_{p} query presentations. The risk head uses N_{r} records of dimension p_{x} and U_{r} full-batch loss/gradient evaluations. With cached features and target masks, fitting costs

$$O\big(N_{\mathrm{sc}}C_{\mathrm{sc}}^{\mathrm{fb}} + U_{\mathrm{c}}(N_{\mathrm{c}}P_\eta + P_{\mathrm{c}}) + N_{\mathrm{p}}(nD + Dr_{\mathrm{p}} + nm) + U_{\mathrm{r}}N_{\mathrm{r}}p_{\mathrm{x}}\big)\,, \tag{42}$$

where n, m are maximum training-pool sizes. Annotation and feature extraction are preprocessing; frozen scorer backbones still incur forward/backward cost. Auxiliary inference storage is $O(nD + nm + F_0B + Be)$ for embeddings, coverage masks, contexts, and exchange proposals, excluding weights, passage text, and neural-model caches.

E Implementation Details

E.1 Baseline Interfaces and Corpus Access

Vanilla and entity-linked retrieval. Vanilla RAG uses the top-four BGE-M3 passages; standalone AppRAG uses the BGE-M3/Qwen3-Embedding-4B union retrieved with the question and its requirements, so initial candidates differ. EntityLinkRAG links mentions to dataset-wide document titles and fuses question/title BM25 retrieval; AppRAG stays within that pool. Unlike this custom baseline, ELERAG (Granata et al., 2026) links to Wikidata and combines embedding similarity with entity overlap.

Graph retrieval. HippoRAG2 and LightRAG index 1,059 prefiltered documents across the three datasets. Each dataset's fixed index deduplicates the union of top-four BGE-M3 passages across sampled questions, using no answers or support labels. Baseline and AppRAG variants share this index; AppRAG selects within the returned pool. LightRAG uses mix mode. Results therefore concern shared prefiltered corpora rather than full-corpus graph retrieval.

ACE adaptation. ACE-RAG combines evidence with a development-frozen playbook shared by both conditions. The baseline uses top-four BGE-M3 passages; the AppRAG variant uses selected evidence. Both then generate with the playbook in the prompt, so ACE-RAG adds a final generation after AppRAG selection instead of returning the saved draft. Appendix A.2 gives the sampling protocol.

Reranking. Both arms share the BGE-M3/Qwen3-Embedding-4B union. RRF-ensemble fuses BGE-reranker-v2-m3 (Chen et al., 2024; Liu et al., 2024), GTE-multilingual-reranker-base, and Qwen3-Reranker-0.6B; RankZephyr/REARANK use released 7B checkpoints. Baselines generate from top-four passages. Named-reranker AppRAG variants admit that reranker's order plus embedding and calibrated rankings only.

E.2 Model Roles and Inference

Qwen3-8B derives fixed atomic requirements from the question alone with greedy decoding and a 192-token limit. A three-member Qwen3-Reranker-4B LoRA ensemble scores evidence. Qwen3-32B checks applicability, coverage, and conflicts through requirement-entailment and misuse-risk views, seeing questions, requirements, and passages without drafts or references. It also compares context–answer pairs without references; generator-scale and replacement-verification diagnostics instead use Qwen3.6-35B-A3B (Qwen Team, 2026) with the same decision rule. Fitted parameters, normalization statistics, all inference models, saved configurations, and the ACE playbook stay frozen; fitting/evaluation separation follows Appendix A.2.

Qwen3-8B generates answers and drafts greedily with thinking disabled and a 64-token limit. Inputs over 16,384 tokens are rejected. Titles and stored passages are serialized in context order, so $B = 4$ limits records, not length. Core AppRAG returns a saved draft; ACE-RAG adds playbook-conditioned generation (Appendix E.1). Answers are saved before scoring, and all full-system configurations retain the risk gate. Alternative contexts can collectively access more than four documents: the main comparison fixes per-context counts, not total compute or auxiliary-model capacity.

Hardware and inference framework. Small-scale validation experiments were run on a single NVIDIA A800 GPU with 80 GB of memory. The full experimental suite was run through an H100 cluster queue, with jobs allocated one to eight GPUs. All experimental LLM backbones, including Qwen3 models from 0.6B to 32B and LLaMA-3.1-8B, used vLLM for inference; Qwen3.6-35B-A3B was also self-hosted with vLLM on H100 GPUs.

E.3 AppRAG Configuration and Training Details

Training data and supervision. All fitting questions come from development collections of HotpotQA, MuSiQue, and 2Wiki, distinct from the main-table evaluation questions. A frozen Qwen3-32B teacher supplies document-level A/IA/IR/U and requirement-coverage annotations; official supporting-document labels provide the support targets. The scorer uses 1,800 questions and 52,611 query–document examples. Each of three members fits two sampling folds (1,200 questions), with the remaining fold reserved for its out-of-fold predictions. The candidate head uses 3,600 development questions and 105,104 document examples, including 979 within-question A/IA pairs. The projector uses a separate 3,600-question development collection with fixed embeddings and applicability/coverage targets. The risk head uses 5,244 proposal records from 1,748 development questions, with three candidate actions per question. No answer-correctness label is used to fit the risk head.

Optimization. The scorer adapts the query and value projections using rank-8 LoRA with scale 16 and a hidden-width-128 multitask head. Each member has about 3.28 million trainable parameters. It is trained for one epoch using AdamW, a microbatch size of four, and four-step gradient accumulation; the adapter and head learning rates are 2×10^{-4} and 10^{-3}, with weight decay 0.01 and gradient clipping at 1.0. Training combines class-weighted cross-entropy for A/IA/IR/U with binary prediction and masked requirement-coverage losses. The three scorer task weights β_{cls}, β_{bin}, and β_{cov} are all one. The candidate head has 1,796 parameters and is fitted for 80 full-batch AdamW updates on frozen features. Its loss equally weights binary cross-entropy and a within-question pairwise logistic ranking term. The residual projector is fitted with AdamW while the embeddings and assessment models remain fixed. It uses two epochs, batches of 16 queries, learning rate 2×10^{-4}, weight decay 0.01, and gradient clipping at 1.0. In Equations 38–40, $\vartheta = 0.05$, $\delta = 0.1$, and $\lambda_{\mathrm{disp}} = 0.1$; the remaining projector loss terms have unit weight. The risk head uses L-BFGS with a strong-Wolfe line search, at most 300 iterations, positive-class-weighted binary cross-entropy, and $\lambda_{\mathrm{risk}} = 10^{-3}$. All feature-normalization statistics are computed from fitting data and then frozen.

Scorer and projector. The candidate head has architecture $18 \to 32 \to 32 \to 4$, with GELU hidden activations and separate sigmoid outputs for relatedness, applicability, inapplicability, and support. Feature means and standard deviations are saved with the fitted head. The applicability weight is the equal average of the binary applicability prediction and A-class probability. The raw IA residual is $p_{\mathrm{IA}}(d)(1 - p_{\mathrm{app}}(d))$; the fixed residual-calibration mapping produces $w_{\mathrm{IA}}(d)$. Here $p_{\mathrm{IA}}(d)$ and $p_{\mathrm{app}}(d)$ are the ensemble binary predictions of inapplicability and applicability. This scalar mapping is distinct from the embedding calibration in § 3.3. The residual query projector has a rank-64 bottleneck, an input bias, and no output bias. Its weighted centers fall back to the query vector when total weight is at most 10^{-8}.

Context construction and risk threshold. At $K = 10$, $B = 4$, alternatives comprise two- and four-document calibrated coverage contexts and a four-document fusion context. Ensemble fusion uses reciprocal ranks with offset 60 over the calibrated, two encoder, and three reranker orders; named-reranker integrations use only their admitted order (Appendix E.1). Fused ordering prioritizes A, U, IR, then IA, followed by conflict flag and fused rank. Anchor-free construction selects the conflict-free A document covering most uncovered requirements, breaking ties by confidence and candidate order. Without added coverage, it selects the next conflict-free A document, then falls back until the budget is filled. Fallbacks consider unselected envelope candidates, preferring non-IA: conflict-

free A first, then incumbent documents, then the remaining ranking. The risk head ρ_ψ uses 60 standardized features $\mathbf{x}$ covering applicability aggregates, verifier labels/conflicts, context size, set overlap, proposal identity, and incumbent-relative changes. All full systems use agreement and $\tau = 0.5$; deletions bypass only the specified stage, and both replacement-verification arms retain the gate. Other K, B values are experiment-specific.

Training objectives and label scope. Appendix D.1 defines the loss terms and the retained projector coverage indices. The risk target is an increase in the fraction of teacher-labeled IA documents, not an increase in answer error or the absolute number of IA documents. The scorer retains an unresolved class in its multiclass task; its binary relatedness target treats unresolved and irrelevant examples as negative. The ensemble members also use different fitting folds, so they are not repeated initializations on identical training examples.

E.4 Metrics for the Overall Ablation

Requirement coverage. For each frozen requirement r_j, an answer-blind judge evaluates the complete evidence context S, allowing support to span several passages. Its label $c_j(q, S)$ is supported, unsupported, or unknown. We report

$$\mathrm{Cov}(q, S) = \frac{1}{m} \sum_{j=1}^{m} \mathbf{1}[c_j(q, S) = \text{supported}]. \tag{43}$$

Unknown and failed judgments remain in the requirement denominator but do not count as confirmed support. This is requirement coverage, not official support-document recall.

Answer-level IA-risk. A separate judgment of $(q, S, \hat{y})$ checks whether an answer assertion contains an explicit entity, relation, time, category, scope, or polarity mismatch localized to the supplied evidence. The result $\ell_{\mathrm{ans}}(q, S, \hat{y})$ is risk, no risk, or unknown, and the reported score is

$$\text{IA-risk}(q, S, \hat{y}) = \mathbf{1}[\ell_{\mathrm{ans}}(q, S, \hat{y}) = \text{risk}]. \tag{44}$$

An unsupported answer or an incorrect benchmark answer is not automatically an IA-risk hit. Unknown and failed judgments remain in the denominator without contributing a confirmed hit; they are not verified negatives. This answer-level measure differs from the document-level IA fraction used to train the risk head and from the context-level IA-hit analysis.

Cov and IA-risk are confirmed lower-bound scores; treating unresolved judgments as positive gives upper bounds. Qwen3.6-35B-A3B supplies these labels. Its prespecified quality checks were not passed, so the columns are descriptive model-based measurements; EM/F1 remain reference-based.

E.5 Main-Text Experiment Protocols

Module ablation. Qwen3-4B scores average both embeddings and $(K, B) \in \{5, 7\} \times \{2, 4\}$ per question, then 600 questions/dataset and equal dataset means. EM/F1 follow Appendix **??**; Qwen3.6-35B-A3B judges Cov/IA-risk. Parentheses show relative changes; prose uses pp.

Generator scale. § 4.3 uses 1,753 shared questions (600 HotpotQA, 600 MuSiQue, 553 2Wiki after exclusions), weighted equally. Six Qwen3 generators from 0.6B to 32B draft with a shared short-answer prompt, greedy decoding, thinking disabled, and a 128-token limit. Contexts have at most four passages, including two-passage alternatives; learned components and the Qwen3.6-35B-A3B comparison judge stay fixed. That judge also scores semantic correctness from question, reference, and answer under two equivalent presentations without method/model identifiers. Only valid agreement fixes a label. Ranges span unknowns, coupled for identical answers in paired contrasts; markers are midpoints.

Support-document ranking. § 4.3 compares raw/calibrated rankings on questions from dataset, excluding validation questions. Pools, document embeddings, and projector are fixed; ranking precedes anchoring, verification, and label joining. Recall counts all annotated supports, including absent ones; complete-support coverage@10 requires all supports in the top ten. Matching uses normalized native paragraph identities and benchmark labels. Questions receive equal weight.

Replacement verification. On the same 180 questions, Qwen3-8B drafts and Qwen3.6-35B-A3B compares context–answer pairs. Ablation changes only verified exchange to unverified anchored

replacement, retaining pools, models, budgets, full-system alternatives, other verifier uses, selection, and $\tau = 0.5$. F1 includes aliases; semantic scores require two-view agreement from the comparison judge.

Context IA exposure. § 4.4 audits saved Qwen3-8B contexts on 90 questions/dataset with Qwen3.6-35B-A3B. Hits require agreement across original/reversed evidence orders and exact source quotes. HotpotQA/MuSiQue use saved judgments; 2Wiki rechecks format/quotation failures on identical inputs with bounded retries and a larger output limit. Each question contributes at most one hit per granularity; unresolved cases remain in the denominator as unknown. Intervals condition on saved judgments, using 10,000 paired bootstrap resamples per dataset with a fixed seed. Displayed differences use rounded rates; statistics use unrounded values. Context totals differ (Vanilla: 360; AppRAG: 294/318/286 passages per dataset), so exposure is not length-matched.

F Prompt Templates

These templates summarize the method and evaluation tasks, with wording and response layouts standardized for presentation rather than copied verbatim from experimental prompts. Bracketed fields denote task inputs.

F.1 Requirements and Applicability Annotation

Requirements are derived from the question before evidence assessment. Offline annotation provides the applicability and coverage targets described in Appendix E.3.

01 Requirement generation — **QUESTION ONLY**

Instruction Identify the facts and conditions needed to answer the question. Express each requirement as one short, independently assessable statement. Preserve the requested entity, relation, time, scope, and answer type. For a multi-hop question, include the intermediate links needed to reach the answer. Use only information stated or required by the question; do not guess missing entities or supply an answer.

Input Question: `[question]`

Response Return a numbered list of atomic requirements. Keep each item concise and retain its identifier for later evidence assessment.

02 Applicability annotation — **OFFLINE TEACHER**

Instruction Assess the passage against the question and its requirements. Identify the reasoning step that the passage could support and check whether its conditions permit that use.

Use **A** for evidence that supports at least one requirement under matching conditions; **IA** for a relevant proposed use with an explicit condition mismatch; **IR** for evidence unrelated to the required reasoning; and **U** when the available information does not resolve the assessment. Partial support can be A. A missing answer or an unverified condition alone is not sufficient for IA.

Inputs Question: `[question]`
Requirements: `[requirement IDs and statements]`
Passage: `[document ID, title, and text]`

Response Report the proposed use, applicability label, covered requirement IDs, and a short evidence-based justification. For IA, identify the mismatched condition; for U, identify what remains unresolved.

F.2 Evidence Verification

The verifier examines the same question, fixed requirements, and candidate passages from two perspectives. Both views are answer-blind. Their predictions inform the context-update rule in § 3.3.

03 Requirement entailment **VERIFICATION VIEW**

Instruction Check what each candidate passage establishes about the fixed requirements. Match the entities, relations, time, and scope before assigning support. A passage may support an intermediate requirement without answering the full question. Distinguish an explicit mismatch from absent information.

Use A, IA, IR, or U according to the applicability definitions. Identify the requirements supported by the passage and flag any explicit conflict with the conditions needed for its proposed use. Base the assessment on the supplied text, without inferring missing links from a possible answer.

Inputs Question: `[question]`
Requirements: `[requirement IDs and statements]`
Candidates: `[document IDs, titles, and passages]`

Response For each document, report its label, confidence, covered requirement IDs, conflict flag, and a brief supporting explanation.

04 Misuse-risk assessment **VERIFICATION VIEW**

Instruction Examine whether using each passage for the requested inference would transfer a fact beyond its stated conditions. Look for explicit mismatches in entity, relation, time, category, scope, or polarity. A shared name or topic does not by itself establish a valid reasoning link.

Assign IA only when you can identify a concrete mismatch in the proposed use. Preserve passages that provide valid partial support. Use U when a necessary condition cannot be established, and IR when the passage does not concern the required reasoning. Do not judge a generated answer.

Inputs Question: `[question]`
Requirements: `[requirement IDs and statements]`
Candidates: `[document IDs, titles, and passages]`

Response For each document, report its applicability label, confidence, covered requirements, and conflict flag. Identify any condition that would be misapplied and the text supporting that judgment.

F.3 Answer Generation and Context–Answer Selection

The generator drafts an answer for each candidate context. The comparison judge then compares the saved context–answer pairs. The agreement and risk rules in § 3.4 determine whether a proposed context is retained.

05 Evidence-conditioned answer generation **GENERATION**

Instruction Answer the question using the supplied evidence. Connect facts across passages when the question requires multiple reasoning steps. Check that the entities and conditions match the question, and distinguish the requested answer from related names or facts appearing in the context.

Return a concise answer in the requested form. Do not include an explanation, a discussion of the retrieval process, or unrelated details.

Inputs Question: `[question]`
Evidence, in context order:
`[document 1: title and passage]`
`[remaining selected documents]`

Response `[short answer]`

06 Context–answer comparison **SELECTION**

Instruction Compare the current context–answer pair with the alternatives. Check whether each context covers the question's requirements and whether its draft follows from evidence used under matching conditions. Consider the verifier assessments alongside the passages; fluent wording alone is not evidence of a better answer.

Keep the current pair unless an alternative is clearly preferable. Use the candidate identifiers when making the decision. Do not infer quality from presentation order, generate a new answer, or combine documents from different candidates.

06 Context–answer comparison (continued)

Inputs Question and requirements: `[question and fixed requirements]`
Current pair: `[context, saved draft, and verifier assessments]`
Alternatives: `[anonymous IDs and corresponding pairs]`

Response Return the identifier of the preferred pair, retaining the current pair when the comparison is unresolved.

The alternatives are presented in both orders using the same comparison task. Agreement is checked across the two judgments before the proposal passes to the learned risk gate.

F.4 Answer and Context Evaluation

These evaluation tasks are separate from evidence selection. The answer judge receives a reference answer; the context audit examines evidence without using a generated answer.

07 Semantic answer evaluation — **ANSWER DIAGNOSTIC**

Instruction Determine whether the predicted answer correctly answers the question according to the reference. Accept equivalent names, aliases, and paraphrases when they denote the same answer. Check the requested relation, granularity, and conditions; topical similarity is not enough.

Judge the answer's meaning rather than its style or explanation length. Use unknown when ambiguity prevents a reliable decision. No method or model identity is supplied.

Inputs Question: `[question]`
Reference: `[reference answer and available aliases]`
Prediction: `[predicted answer]`

Response Return correct, incorrect, or unknown, with a short justification.

08 Inapplicable-evidence audit — **CONTEXT DIAGNOSTIC**

Instruction Inspect the final evidence context for relevant passages whose proposed use conflicts with the question's requirements. Distinguish entity-level mismatches from other evidence-level condition mismatches. A passage that supports only part of the reasoning is not automatically inapplicable.

For each detected mismatch, identify the document, the affected condition, and an exact supporting quotation. If the text does not establish a mismatch or a reliable negative judgment, retain an unresolved outcome. Evaluate the supplied context rather than whether an answer adopted it.

Inputs Question: `[question]`
Requirements: `[fixed requirements]`
Context: `[document IDs, titles, and passages]`

Response Report each mismatch with its type, document ID, and source quotation. Distinguish detected mismatches, no detected mismatch, and unresolved cases.

Semantic evaluation uses equivalent presentations of the same answer. The context audit uses original and reversed evidence orders, with quotation validation. Agreement and unresolved outcomes are handled by the protocols in Appendix E.5.

F.5 Coverage and Answer-Level Risk Diagnostics

These tasks supply the diagnostic Cov and IA-risk measurements defined in Appendix E.4. Coverage is assessed from the evidence; answer-level risk additionally requires the generated answer.

09 Requirement coverage — **COVERAGE DIAGNOSTIC**

Instruction Assess each fixed requirement using the complete evidence context. Allow support to span multiple passages when the connecting facts and conditions are established. Do not require one passage to answer the entire question, and do not fill missing reasoning links from an assumed answer.

Assign supported when the supplied evidence establishes the requirement, unsupported when it does not, and unknown when the available information does not permit a reliable assessment.

09 Requirement coverage (continued)

Inputs Question: `[question]`
Requirements: `[requirement IDs and statements]`
Context: `[document IDs, titles, and passages]`

Response For each requirement ID, return supported, unsupported, or unknown, and identify the evidence for the judgment.

10 Answer-level applicability risk **RISK DIAGNOSTIC**

Instruction Check whether an assertion in the answer uses the supplied evidence under an explicitly mismatched entity, relation, time, category, scope, or polarity condition. Locate both the affected assertion and the evidence that establishes the mismatch.

An unsupported assertion or an incorrect answer alone does not establish applicability risk. Distinguish a confirmed mismatch from missing evidence and unresolved interpretation. Do not use a reference answer to decide whether evidence conditions match.

Inputs Question: `[question]`
Context: `[document IDs, titles, and passages]`
Answer: `[generated answer]`

Response Return risk, no risk, or unknown. For risk, identify the assertion, the mismatched condition, and the supporting source text.

G LIMITATIONS

AppRAG depends on the retrieved candidate pool and predicted requirements and applicability. Shared model errors can affect verification and draft comparison. Evaluation covers three multi-hop QA benchmarks; additional model calls increase inference cost.