CORAL: Claim Organization via Relation-Aware Linking for Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

Retrieved passages in retrieval-augmented generation (RAG) systems often contain redundant facts, distracting information, and multi-hop clues scattered across documents. Existing methods address this noise by compressing passages into free-form summary notes, but such summaries leave cross-document structure implicit: filtering, deduplication, and multi-hop linking are folded into a single generation step with no explicit mechanism to perform them. The downstream generator must then recover these reasoning chains on its own, which frequently leads to errors on complex multi-hop queries. We propose CORAL, a RAG system that restructures the pipeline into an iterative retrieve-organize-reason loop operating at the atomic claim level, where the agent autonomously decides when to search, what to organize, and when to answer. After each retrieval step, the model decomposes passages into source-grounded claims, then organizes them through relation-specific resolutions: filtering non-contributory claims, merging overlapping ones across documents, chaining bridging claims to form multi-hop reasoning paths, and retaining isolated claims. Per-claim notes encode each claim's relational role, serving as a compact organized representation that guides subsequent retrieval or answer generation. We optimize the entire system end-to-end via reinforcement learning, eliminating the need for gold intermediate annotations. Across seven QA benchmarks, CORAL achieves the best average performance and yields statistically significant gains on all multi-hop benchmarks, improving over the strongest average baseline by 4.2 Exact Match points. Our code is anonymously available at https://anonymous.4open.science/r/CORAL-C9B8.

1 Introduction

Retrieval-augmented generation (RAG) has emerged as a dominant paradigm for reducing hallucination in large language models (LLMs) by grounding generation in external evidence (Lewis et al., 2020; Guu et al., 2020; Gao et al., 2023). A fundamental challenge is that retrieved evidence is often noisy and poorly organized: a typical retrieval set contains redundant statements of the same fact, partially relevant background, and topically related yet misleading distractors across multiple documents. This burden becomes especially acute for questions that require connecting information across documents (Yang et al., 2018; Trivedi et al., 2023).

Recent work addresses this issue by introducing an intermediate evidence structuring step between retrieval and generation. For instance, Chain-of-Note (Yu et al., 2024) reads retrieved documents sequentially and writes reading notes that judge each document's relevance while extracting useful information. EviNote-RAG (Dai et al., 2025) trains the model to produce Supportive-Evidence Notes that compress retrieved content into notes with explicit key and uncertainty annotations. Although these methods reduce noise, they rely on free-form summarization that couples filtering, deduplication, and cross-document linking into a single text generation step. The resulting summaries leave cross-document structure implicit, so the downstream generator must still recover how pieces of evidence relate to one another. This limitation is particularly pronounced on multi-hop questions, where the reasoning path itself must be reconstructed from the summary.

We argue that effective evidence organization requires operating at the level of atomic claims. A single passage often conflates facts that serve entirely different roles with respect to a given query, such as a person's biography mixing authorship, birthplace, and career facts into a single paragraph. Summary-based methods lack the discrete units needed for selective filtering, deduplication, and cross-document linking. Our claim decomposition (§3.2) addresses this by providing individually

addressable evidence units. While post-hoc evaluation methods extract atomic facts from generated text to verify factuality (Min et al., 2023; Wei et al., 2024), we leverage claim-level decomposition proactively to structure raw retrieved evidence for downstream reasoning. We propose CORAL, a system that restructures the RAG pipeline into an iterative retrieve-organize-reason loop in which an autonomous agent controls the search–organize–answer cycle. CORAL decomposes retrieved text into atomic claims. It applies relation-specific resolutions to explicitly structure the retrieved evidence, addressing dependencies that passage-level notes leave implicit: filtering out distractors, merging redundant facts across documents, chaining bridging claims to form multi-hop reasoning paths, and retaining isolated claims. Finally, it attaches a specific natural language note to each retained claim to encode its relational role. These per-claim notes serve as a structured, dynamic memory state, allowing the agent to either formulate a new search query if evidence is lacking, or proceed to generate the final answer.

We train the entire system end-to-end via reinforcement learning with a composite reward combining answer correctness, grounding, and format constraints, requiring no gold intermediate annotations.

Our contributions are as follows:

1. We propose CORAL, an agentic RAG system that performs query-conditioned evidence organization at the atomic claim level within an iterative retrieve-organize-reason loop. CORAL uses atomic claims as individually addressable evidence units and applies relation-specific resolutions (filtering, merging, chaining, and retaining) to explicitly structure retrieved evidence before answer generation.
2. We introduce per-claim notes as a dynamic memory state that persists across retrieval turns, encoding each claim's relational role and guiding subsequent search and reasoning.
3. We empirically validate CORAL across seven QA benchmarks. Trained end-to-end via reinforcement learning, CORAL outperforms strong baselines, with especially large gains on multi-hop reasoning tasks (+4.2 EM over the strongest average baseline on multi-hop benchmarks) and a 2.4-point overall average EM improvement across all seven datasets.

2 Related Work

2.1 Retrieval-Augmented Generation

RAG systems combine a retriever with a generator to ground responses in external knowledge (Lewis et al., 2020; Guu et al., 2020; Karpukhin et al., 2020; Ram et al., 2023). Standard pipelines retrieve top-k passages and concatenate them as context, but typical retrieval sets contain redundant statements, partially relevant background, and topically related distractors that propagate to the final answer (Shi et al., 2023; Yoran et al., 2024). Adaptive and iterative variants (Asai et al., 2024; Jiang et al., 2023; Trivedi et al., 2023) dynamically decide when and what to retrieve, yet still reason over the resulting passages in their raw form, leaving the burden of organizing evidence to the generator.

2.2 Evidence Structuring in RAG

To mitigate retrieval noise, recent approaches interleave an evidence processing step before generation. One line of work generates intermediate reading notes that judge document relevance and summarize key information (Yu et al., 2024; Dai et al., 2025). While effective at reducing noise, these approaches produce unstructured natural language summaries where inter-document dependencies remain implicit, requiring the final generator to infer how individual evidence pieces relate to one another. A complementary line of work prebuilds graph-based or hierarchical structures over the entire corpus (Edge et al., 2024; Sarthi et al., 2024; Gutiérrez et al., 2024, 2025). These methods target the retrieval stage, constructing offline, query-agnostic indices at the passage or entity level to broaden cross-document connectivity. However, they do not address how the generator should consume and reason over the retrieved results. For a given query, they cannot prescribe which pieces of evidence should be merged as duplicates, chained as logical bridges, or retained as independent support. CORAL operates at a different point in the pipeline: after retrieval, it decomposes the returned passages into atomic claims and assigns each claim a query-conditioned relational role that prescribes a specific resolution, explicitly structuring the evidence for the downstream generator.

2.3 Claim Decomposition for Verification and Retrieval

Atomic claim decomposition is widely used for post-hoc factuality evaluation, where claims are ex-

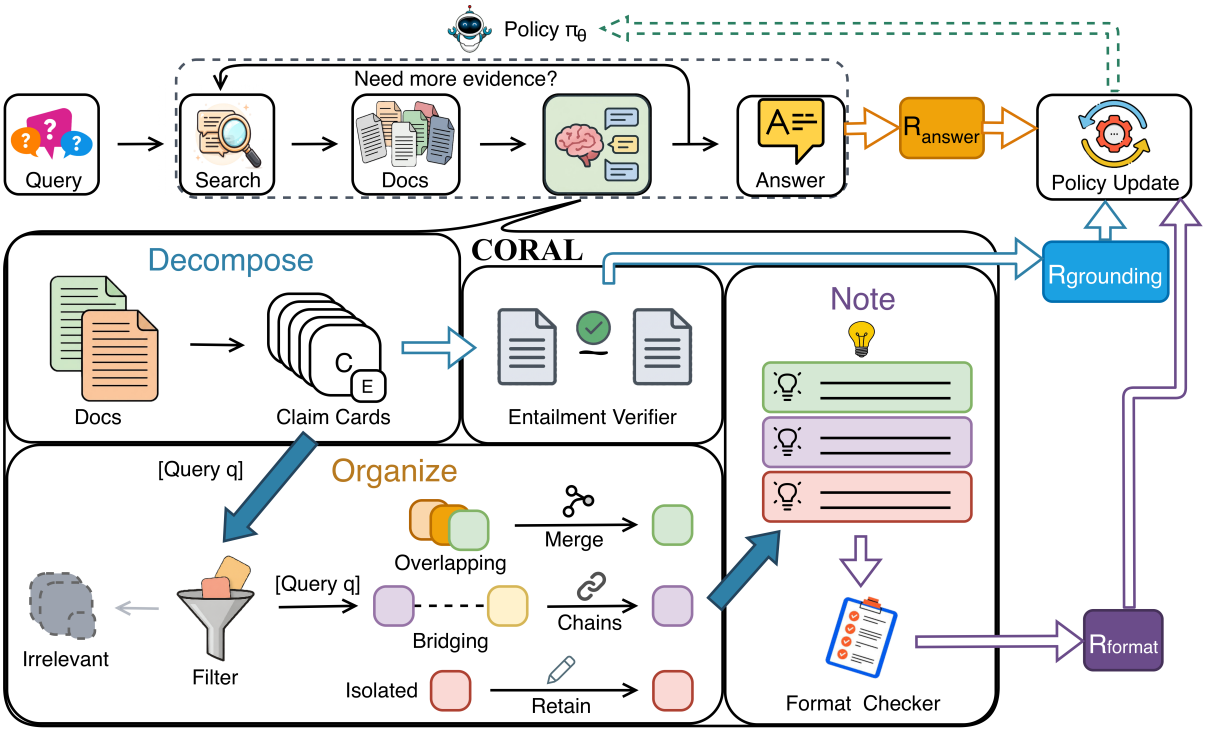

Figure 1: Overview of the CORAL system. Top: An iterative retrieval loop where the system gathers documents, processes them via CORAL, and optionally issues new queries before generating the final answer. Bottom: Inside CORAL, raw documents are decomposed into atomic claims. The claims are then filtered for relevance and organized by resolving overlapping, bridging, and isolated relational states. Finally, they are converted into structured notes. The entire trajectory is optimized via GRPO using three reward signals: answer reward (R_{answer}), grounding reward ($R_{\text{grounding}}$), and format reward (R_{format}).

tracted from generated text and checked against a reference (Min et al., 2023; Wei et al., 2024; Kamoi et al., 2023). A separate line of work uses propositions or atomic facts as offline knowledge units, either replacing passage-level chunks in the retrieval index (Chen et al., 2024) or arranging them into a corpus-level proposition graph for multi-hop search (Wang and Han, 2025). CORAL differs along both axes: it decomposes claims at inference time from retrieved passages, and treats them as nodes whose query-conditioned relational state determines a specific resolution action prior to generation.

2.4 Reinforcement Learning for RAG

Reinforcement learning from outcome signals has proven effective for improving LLM reasoning capabilities (DeepSeek-AI, 2025; Shao et al., 2024). Within RAG, recent agentic systems use RL to interleave search and reasoning over raw passages (Jin et al., 2025; Song et al., 2025), and EviNote-RAG (Dai et al., 2025) optimizes free-form evidence summaries for downstream answer accuracy. CORAL shows that the same outcome-driven signal, regularized with grounding and format rewards, can additionally induce structured, claim-level evidence organization without gold intermediate annotations.

3 Method: CORAL

3.1 Overview

CORAL inserts an explicit claim-level organization stage into an iterative retrieval loop. After each retrieval, the model decomposes the retrieved passages into atomic, source-grounded claims. It then organizes these claims by conservatively excluding non-contributory ones and identifying query-conditioned relational states among the retained claims. Overlapping claims are merged to reduce redundancy, bridging claims are chained to form directed evidence paths for reasoning, and isolated claims are retained as standalone evidence. Finally, the organized claims are converted into lightweight natural language notes. The full pipeline is illustrated in Figure 1.

Three properties distinguish this organization from free-form note generation. First, each claim is an individually addressable evidence unit with

explicit provenance, enabling selective operations at sub-passage granularity. Second, each relational state prescribes exactly one resolution action: overlapping → merge, bridging → chain, isolated → retain. This one-to-one mapping eliminates ambiguity in how evidence should be processed once the relational state is identified. Third, notes from round t are injected as structured input to round $t{+}1$'s organize stage, which allows cross-round relationship discovery and incremental evidence accumulation.

These notes replace raw passages in the model context and serve as a dynamic memory state across retrieval turns: in each subsequent round, only the note buffer and current-turn passages are provided as input. If the model determines that evidence is insufficient, it formulates a targeted follow-up search conditioned on existing notes. Once sufficient evidence is gathered, it generates the final answer.

3.2 Claim Decomposition

Given a query q and a set of retrieved passages $D = \{d_1, \ldots, d_k\}$, the first step of CORAL maps the raw text into a set of atomic claims $\mathcal{C} = \{(c_i, s_i)\}_{i=1}^{n}$, where c_i denotes the textual proposition and $s_i \in D$ specifies its provenance.

An atomic claim must be (i) **atomic**: exactly one factual proposition; (ii) **self-contained**: all pronouns and implicit references disambiguated; (iii) **source-grounded**: cited to its originating document; and (iv) **verifiable**: independently checkable as true or false, excluding opinions or meta-information. The policy model π_θ autoregressively extracts claims satisfying these criteria, as illustrated in Figure 2.

This sub-passage granularity is a prerequisite for the subsequent organization stage, allowing selective filtering and providing the discrete units for cross-document relational reasoning.

3.3 Relation-Aware Claim Organization

Given the claim set $\mathcal{C}$ and the query q, the second step organizes claims into a compact, non-redundant evidence structure through three stages: conservative relevance triage, relationship discovery, and structural resolution. In multi-turn settings, the organization stage additionally receives accumulated notes from previous rounds. These are treated as existing claims during relationship discovery, so that a newly retrieved claim can be identified as overlapping or bridging with a previously established note.

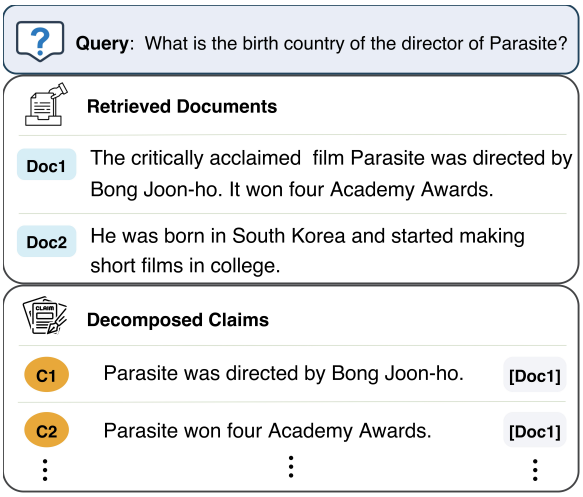

Figure 2: An example of claim decomposition. The model extracts atomic claims from raw passages and disambiguates coreferences to ensure each claim is strictly self-contained.

Relevance Triage. To eliminate retrieval noise, the policy model π_θ performs a conservative utility judgment over $\mathcal{C}$, conditioned on both q and the global context of the claim set. The model assesses whether a claim possesses direct answer-bearing utility or indirect value as a latent bridging step in a multi-hop reasoning path. The triage is conservative: a claim is excluded only when it is clearly non-contributory along both dimensions. When uncertain, the model retains the claim to avoid premature deletion of bridging evidence.

Query-Conditioned Relationship Discovery. Following triage, the model identifies semantic dependencies among the retained claims. To optimize for factoid QA, we distill complex discourse relations (Radev, 2000) into a minimal, actionable taxonomy consisting of three relational states:

- **Overlapping**: Claims that provide semantically equivalent or redundant information with respect to q.
- **Bridging**: Claims where one proposition introduces an entity or premise required to interpret the other, forming a logical dependency.
- **Isolated**: Claims that are relevant to the query but possess no structural dependencies with other claims.

These three states capture the main structures needed for cross-document factoid QA and align with established multi-hop reasoning typologies (Yang et al., 2018; Trivedi et al., 2022). Once as-

signed, each maps to a fixed resolution operation. We omit a contradictory state because genuine factual contradictions are rare in Wikipedia-based factoid QA (Semnani et al., 2025). Apparent conflicts typically arise from entity ambiguity or specificity differences and can be resolved through relevance triage or as overlapping evidence. By avoiding the sparsity and ambiguity of broader relation schemas (Radev et al., 2004; Zhang et al., 2002), we retain only the relations needed for downstream evidence resolution. The reliability of the predicted states is evaluated in §4.4.

A key property of these relational states is that they are dynamically conditioned on the query. For instance, the claims "*Parasite was directed by Bong Joon-ho*" and "*Bong Joon-ho was born in South Korea*" exhibit a bridging state for the query "*Where was the director of Parasite born?*", but for the query "*Who directed Parasite?*" the first claim directly answers the question while the second is excluded as non-contributory.

Structural Resolution. Finally, the model resolves the triage decisions and identified relational states through type-specific actions, producing a refined evidence structure: (1) **Exclude** removes claims deemed non-contributory during relevance triage; (2) **Merge** fuses overlapping claims into a single representative claim with aggregated provenance, reducing redundancy; (3) **Chain** connects bridging claims via directed logical dependencies, forming explicit reasoning paths; and (4) **Retain** preserves isolated claims as independent units. This process explicitly resolves inter-claim dependencies prior to the final reasoning step.

3.4 Claim-Level Note Generation

Given the refined evidence structure and query q, the third step translates the discrete organization into textual annotations that can be directly consumed by the language model.

We augment each claim surviving the organization stage with a natural language note that encodes its relational role in answering the query. Each note specifies: (1) the claim's functional role; (2) its source strength (number of corroborating documents); (3) its connections to other claims within a reasoning chain; and (4) a usage instruction describing how the downstream generator should employ it. The assigned roles depend dynamically on the query context. In single-hop scenarios, a note may designate a claim as the direct answer. In multi-hop scenarios, notes may instruct the generator to use claims as intermediate entity identifiers, logical bridge components, or supporting evidence. Figure 3 illustrates an example.

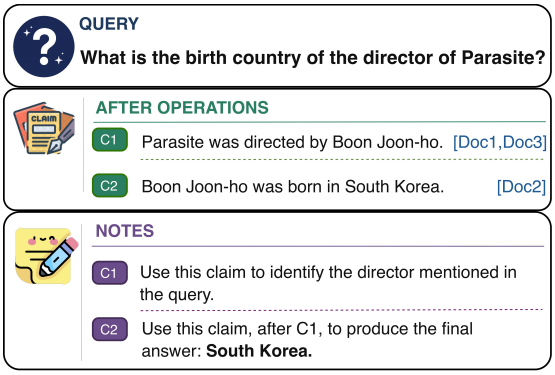

Figure 3: Simplified example of per-claim notes. Each note encodes the relational role of the claim and prescribes its use in the final answer generation phase.

The annotated claims replace raw passages as generator context, with notes carrying forward across retrieval turns rather than accumulating raw passages. After note generation, the model reviews the accumulated notes and decides whether to issue a follow-up search or generate the final answer. Specifically, if the notes reveal an incomplete reasoning chain (e.g., an entry-point claim with no answer-bearing follow-up), the model formulates a targeted query to fill the gap; otherwise, it synthesizes the notes into the final answer. This decision is learned jointly with the rest of the pipeline via Reinforcement Learning.

3.5 End-to-End RL Training

CORAL is formulated as an agentic RAG policy that interleaves reasoning, retrieval, evidence organization, and answer generation. For each query, the unified policy model π_θ autoregressively generates a complete trajectory τ, including search query formulation, passage retrieval, claim decomposition, evidence organization, note generation, search-or-answer decisions, and final answer prediction.

Reward Design. We optimize π_θ with a composite trajectory-level reward that combines answer correctness with lightweight structural constraints:

$$R(\tau) = \lambda_1 R_{\text{answer}} + \lambda_2 R_{\text{grounding}} + \lambda_3 R_{\text{format}}. \quad (1)$$

Here, $R_{\text{answer}} \in \{0, 1\}$ indicates whether the final answer matches the gold answer under exact match, while $R_{\text{grounding}}$ encourages faithful claim

decomposition by measuring whether each claim c_i is entailed by its attributed source passage s_i using a frozen Natural Language Inference (NLI) model:

$$R_{\text{grounding}} = \frac{1}{n} \sum_{i=1}^{n} P_{\text{NLI}}(\text{entail} \mid s_i, c_i), \quad (2)$$

where n is the total number of decomposed claims across all retrieval rounds in the trajectory. Since $R_{\text{grounding}}$ is averaged over generated claims, a potential concern is that the model might generate fewer claims to inflate this score. In practice, R_{answer} ($\lambda_1 = 1.0$) provides strong implicit coverage pressure: omitting claims that are necessary for multi-hop reasoning leads to incorrect answers, incurring a penalty that far outweighs any grounding reward gain. We verify empirically that claim counts do not decrease during training (Appendix R).

R_{format} is a binary reward that checks for required trajectory markers, including search actions, claim decomposition, relational state prediction, note generation, and final answer generation. Together, these three rewards encourage trajectories that are structurally well-formed, grounded in the retrieved evidence, and correct in their final answer. Note that $R_{\text{grounding}}$ is applied to decomposed source-grounded claims; the subsequent note generation stage is constrained to annotate retained or merged claims rather than introduce new unsupported propositions, so grounding supervision propagates transitively to the generator's input.

Optimization via GRPO. We adopt critic-free GRPO (Shao et al., 2024) to avoid maintaining a separate value model. For each query q, we sample G trajectories from π_θ and normalize rewards within the group:

$$\hat{A}_i = \frac{R(\tau_i) - \text{mean}(\{R(\tau_j)\}_{j=1}^{G})}{\text{std}(\{R(\tau_j)\}_{j=1}^{G})}. \quad (3)$$

These relative advantages are subsequently used to weight the policy gradients across all tokens within the trajectory. To prevent reward hacking and maintain language fluency, we apply a standard Kullback-Leibler (KL) divergence penalty against a frozen reference policy. This formulation allows the model to jointly optimize its retrieval triggers, evidence organization strategies, and downstream reasoning logic.

4 Experiments

4.1 Setup

We evaluate CORAL on seven QA benchmarks, encompassing both multi-hop (HotpotQA, 2WikiMultiHopQA, MuSiQue, Bamboogle) and single-hop (Natural Questions, TriviaQA, PopQA) reasoning tasks. Our baselines span multiple RAG paradigms, including standard RAG, adaptive RAG, prompt-based agentic methods, and RL-based agentic methods with and without intermediate evidence notes. We use Qwen2.5-7B-Instruct (Qwen Team, 2024) as the base model for CORAL and all prompt-based baselines. For RL-based baselines, we use their officially released checkpoints evaluated under the canonical retrieval configuration introduced by Search-R1 (Jin et al., 2025). Performance is measured using Exact Match (EM) and token-level F1. Additional experimental details are provided in Appendix A.

4.2 Main Results

Table 1 presents the main results across seven benchmarks.

CORAL achieves the best overall performance, with an average EM of 50.3 and an average F1 of 56.4. It improves over the best prior baseline by 2.4 EM points on average, and outperforms R1-Searcher, the strongest RL-based method that does not use intermediate evidence notes, by 9.2 EM points, corresponding to a 22.4% relative gain.

The gains are concentrated on multi-hop benchmarks, where CORAL outperforms all baselines on HotpotQA, 2WikiMultiHopQA, MuSiQue, and Bamboogle, confirming the value of explicit claim-level organization for exposing cross-document reasoning paths.

We also note that several baselines exhibit high variance across datasets (e.g., ReARTeR achieves 53.4 EM on 2Wiki but only 9.6 on Bamboogle), whereas CORAL maintains consistently strong performance across all multi-hop benchmarks.

On single-hop benchmarks, CORAL remains competitive with the best baselines, with near-parity to Search-R1 on PopQA, suggesting that claim-level organization does not substantially hurt direct factual retrieval.

All reported CORAL results are averaged over 3 training seeds. Paired bootstrap tests (10K resamples) confirm that the multi-hop improvements are statistically significant ($p < 0.01$) on all four benchmarks, while single-hop differences are

Method	Multi-hop QA								Single-hop QA						Avg	
	HotpotQA		2Wiki		MuSiQue		Bamboogle		NQ		TriviaQA		PopQA			
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Prompting-based																
Direct	21.6	24.4	22.8	26.4	7.4	8.5	21.6	22.1	13.4	19.8	29.8	32.1	15.0	17.0	18.8	21.5
Naive RAG	33.0	37.1	23.2	24.4	9.4	10.0	24.0	25.4	37.6	42.0	44.2	45.6	45.8	47.9	31.0	33.2
IRCoT	39.2	43.5	41.7	49.2	10.2	19.2	11.2	24.5	47.0	51.2	41.8	43.2	28.7	32.2	31.4	37.6
Self-RAG	40.2	44.8	39.1	43.2	18.3	19.2	25.6	35.1	46.5	50.8	43.6	45.1	31.4	33.2	35.0	38.8
Search-o1	26.8	33.0	27.2	28.6	14.0	16.8	32.8	35.8	31.0	34.5	56.6	58.9	33.6	36.9	31.7	34.9
RL-based																
R1-Searcher	45.1	49.8	44.6	49.1	21.4	22.8	17.6	20.1	49.2	53.8	68.8	73.1	41.3	42.7	41.1	44.5
Search-R1	42.0	46.4	24.4	28.0	18.4	27.4	32.0	37.7	50.8	55.0	69.4	75.4	**48.2**	**49.8**	40.7	45.7
ReARTeR	46.5	51.2	53.4	55.4	23.7	29.6	9.6	11.9	50.2	54.5	50.6	54.8	42.2	43.2	39.5	42.9
EviNote-RAG	49.0	55.7	49.4	53.6	24.0	33.6	42.4	52.8	**52.4**	56.3	70.3	76.5	48.0	49.1	47.9	53.9
CORAL	**53.0**	**59.4**	**54.0**	**58.0**	**28.2**	**37.0**	**46.4**	**56.0**	**52.4**	**56.8**	**70.8**	**77.8**	47.6	49.6	**50.3**	**56.4**

Table 1: Main results on seven QA benchmarks. CORAL and prompt-based baselines use Qwen2.5-7B-Instruct as the base model; RL baselines use their officially released checkpoints evaluated under the canonical E5 retrieval setup. CORAL results are averaged over 3 training seeds (per-dataset standard deviations in Appendix E). **Bold**: best; underline: second best.

largely not significant, consistent with gains being concentrated on multi-hop reasoning. Per-dataset p-values are reported in Appendix L.

A finer-grained breakdown by hop count (Appendix F) shows gains scaling monotonically: +0.8 EM on single-hop, +3.8 on 2-hop, and +5.2 on 3+ hop questions, confirming that bridging chain construction becomes progressively more important as reasoning paths grow longer.

4.3 Ablation Study

To isolate the contribution of each component, we conduct an ablation study by training variants with specific modules removed from the pipeline. Table 2 reports the results averaged across all seven benchmarks (per-dataset breakdown in Appendix D).

Setting	Avg EM	Avg F1
CORAL (Full)	**50.3**	**56.4**
− Notes	47.0	52.6
− Organization	44.0	49.8
− Organization − Notes	41.6	46.8
− $R_{\text{grounding}}$	48.6	54.2
Prompt-only (no RL)	25.2	29.8

Table 2: Ablation study. The top block removes pipeline components from the full model; the middle block tests reward design; the bottom block removes RL training entirely. Scores are averaged over all seven benchmarks.

Each ablation variant retains claim decomposition but removes specific downstream modules. − **Notes** performs the full organization but skips per-claim note generation. − **Organization** decomposes claims but skips triage, relation discovery, and structural resolution entirely. The flat claim list is annotated with per-claim notes that lack relational context and then passed to the generator. − **Organization** − **Notes** retains only claim decomposition. The raw claim list is passed directly to the generator.

The results show that both explicit notes and evidence organization are important, with organization contributing more substantially. Disabling note generation lowers EM by 3.3 points, indicating that usage annotations help the generator determine how each claim should support the final answer. Disabling evidence organization causes a larger drop of 6.3 EM points, suggesting that filtering, merging, and chaining claims are especially important for multi-evidence reasoning.

The two modules are complementary but not simply additive. When both are disabled, EM decreases by 8.7 points to 41.6, slightly smaller than the sum of the individual drops, suggesting that notes are most effective when built on well-organized claims. Comparing the decomposition-only variant (41.6) against Naive RAG (31.0) isolates the effect of RL-trained claim decomposition, while comparing the full model (50.3) against this variant (41.6) isolates the combined contribution of organization and notes.

The prompt-only baseline performs below Naive RAG (25.2 vs. 31.0 EM) because the untrained

model expends most of its generation budget on poorly-executed intermediate structures, confirming that effective evidence organization requires RL training rather than prompting alone.

The grounding reward mainly affects faithfulness. Removing $R_{\text{grounding}}$ reduces average EM by 1.7 points, but Section 4.4 shows that it triples the hallucination rate. This indicates that grounding supervision is crucial for keeping intermediate claims faithful to retrieved evidence, even when its effect on final-answer metrics is moderate.

4.4 Analysis

Claim faithfulness. To verify the faithfulness of decomposed claims, we randomly sample 200 claims from each of three settings (Full, $-R_{\text{grounding}}$, Prompt-only; 600 total) and have three annotators independently judge whether each claim is entailed by its cited source passage (average pairwise Cohen's $\kappa = 0.82$, majority vote as gold). Table 3 reports the results.

Setting	Faithful %	Hallucinated %
CORAL (Full)	94.2	5.8
$-R_{\text{grounding}}$	81.6	18.4
Prompt-only	71.3	28.7

Table 3: Claim faithfulness evaluation on 200 manually annotated claims per setting (600 total). $R_{\text{grounding}}$ significantly reduces hallucinated claims.

The full model achieves 94.2% faithfulness. Removing $R_{\text{grounding}}$ increases hallucination from 5.8% to 18.4%, as the model occasionally generates plausible but unsourced claims that happen to yield correct answers, a form of reward hacking that the entailment-based reward effectively suppresses. The prompt-only baseline exhibits the highest hallucination rate (28.7%), confirming that faithful decomposition requires explicit grounding supervision rather than instruction alone.

Relational state prediction quality. To assess intermediate relational state judgments, we conduct a two-part human evaluation: for *precision*, we sample 50 predictions per type (150 total) and have three annotators verify correctness; for *recall*, we sample 320 claim pairs from 80 queries and have annotators identify ground-truth relations (average pairwise Cohen's $\kappa = 0.79$; full details in Appendix M). Table 4 reports the results.

Overlapping judgments achieve the highest precision (92.0%), aided by strong lexical cues, while bridging precision is lower (82.0%), with errors primarily involving direction reversal or missing intermediate steps (Appendix O). Recall is consistently lower than precision across all types, indicating a conservative bias: the model misses valid relations rather than hallucinating spurious ones, consistent with the training objective where false merges or incorrect chains are more harmful than mild redundancy from undetected overlap. Overall, the outcome-driven RL signal induces reasonably accurate relational state prediction without explicit relation-level supervision. Additional analyses are provided in the Appendix.

Judgment Type	N	Prec.	Rec.	F1
Irrelevant	50 / 320	88.0	79.4	83.5
Overlapping	50 / 320	92.0	84.6	88.1
Bridging	50 / 320	82.0	72.8	77.1

Table 4: Precision, recall, and F1 of intermediate organization judgments: the Relevance Triage decision (Irrelevant) and the two explicitly predicted relational states (Overlapping, Bridging). Isolated is omitted as it is the residual label for retained claims not assigned to any pair. N: precision samples / total pairs annotated for recall.

5 Conclusion

Taken together, our results suggest that evidence organization in RAG is better framed as a structured claim-level decision problem than as passage-level summarization. Across seven QA benchmarks, our experiments show that CORAL is particularly effective on multi-hop and distractor-heavy questions, where explicit filtering, merging, and chaining help recover cross-document reasoning paths. At the same time, it remains competitive in single-hop settings, indicating that structured claim organization improves complex reasoning without sacrificing simpler factual retrieval. The results further show that outcome-based RL, regularized with lightweight grounding and format constraints, can induce this organization behavior without gold intermediate annotations. Beyond the accuracy gains, the resulting claim structures provide a transparent intermediate state for inspecting how evidence is selected and connected, suggesting a promising direction for future RAG systems that require both accuracy and interpretability.

Limitations

CORAL incurs greater computational cost than standard RAG due to its structured intermediate generation process, leading to higher latency that may limit its deployment in latency-sensitive settings. Our experiments are limited to English factoid QA benchmarks, and CORAL's effectiveness on other languages and task types (e.g., open-ended generation and summarization) remains to be validated. The current relational taxonomy is intentionally scoped to factoid QA. Extending CORAL to domains where genuine factual contradictions or temporal conflicts are common would likely require additional relational states and corresponding resolution mechanisms, which we leave to future work. The quality of claim decomposition depends on the base model's ability to perform faithful decomposition. Smaller or weaker models may struggle to produce faithful atomic claims even with grounding reward supervision. Finally, although we follow prior work by training on NQ and HotpotQA and evaluating on disjoint test splits, this setup may still benefit from dataset-level familiarity.

Ethical Considerations

Our work presents an agentic Retrieval-Augmented Generation (RAG) system aimed at improving the faithfulness and reasoning capabilities of language models. Although claim-level decomposition and grounding rewards substantially reduce hallucinations, the system remains dependent on the quality and neutrality of the retrieved source documents. If the underlying corpus contains biased, toxic, or factually incorrect information, the model may still reproduce these issues. Moreover, as with any LLM-based system, there remains a residual risk of generating plausible yet incorrect statements. We therefore recommend careful oversight and appropriate human-in-the-loop verification when deploying such systems in high-stakes settings such as medical or legal question answering. All models and datasets used in this research are publicly available and were used in accordance with their respective licenses.

References

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

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024. Dense X retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Yuqin Dai, Guoqing Wang, Yuan Wang, Kairan Dou, Kaichen Zhou, Zhanwei Zhang, Shuo Yang, Fei Tang, Jun Yin, Pengyu Zeng, Zhenzhe Ying, Can Yi, Changhua Meng, Yuchen Zhou, Yongliang Shen, and Shuai Lu. 2025. EviNote-RAG: Enhancing RAG models via answer-supportive evidence notes. *arXiv preprint arXiv:2509.00877*.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From RAG to memory: Non-parametric continual learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. REALM: Retrieval-augmented language model pre-training. In *International Conference on Machine Learning*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. WiCE: Real-world entailment for claims in wikipedia. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5420–5438.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

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

Dragomir Radev. 2000. A common theory of information fusion from multiple text sources step one: Cross-document structure. In *1st SIGdial Workshop on Discourse and Dialogue*, pages 74–83.

Dragomir Radev, Jahna Otterbacher, and Zhu Zhang. 2004. CST bank: A corpus for the study of cross-document structural relationships. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*.

Sina Semnani, Jirayu Burapacheep, Arpandeep Khatua, Thanawan Atchariyachanvanit, Zheng Wang, and Monica Lam. 2025. Detecting corpus-level knowledge inconsistencies in Wikipedia with large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34839–34866.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. HybridFlow: A flexible and efficient RLHF framework. *arXiv preprint arXiv:2409.19256*.

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

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-Searcher: Incentivizing the search capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2503.05592*.

Zhongxiang Sun, Qipeng Wang, Weijie Yu, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Song Yang, and Han Li. 2025. ReARTeR: Retrieval-augmented reasoning with trustworthy process rewarding. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1251–1261.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Jingjin Wang and Jiawei Han. 2025. PropRAG: Guiding retrieval with beam search over proposition paths. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. Long-form factuality in large language models. In *Advances in Neural Information Processing Systems*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

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

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. 2024. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Zhu Zhang, Sasha Blair-Goldensohn, and Dragomir R. Radev. 2002. Towards CST-enhanced summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

A Detailed Experimental Setup

Datasets. We evaluate on seven QA benchmarks spanning different reasoning requirements. **Multi-hop QA**: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023) require composing multiple evidence pieces across documents. **Single-hop QA**: Natural Questions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) verify that the method does not over-engineer simple queries. PopQA (Mallen et al., 2023) further tests robustness on rare entities where parametric memory is less reliable. Following prior work (Jin et al., 2025; Dai et al., 2025), we train on the combined training sets of NQ and HotpotQA, while the remaining five datasets serve as out-of-domain evaluation. We randomly sample 500 examples from each test or validation set for evaluation using a fixed random seed (seed = 42); for Bamboogle, we use the full validation set of 125 examples. This sample size follows the evaluation protocol of Search-R1 (Jin et al., 2025) and EviNote-RAG (Dai et al., 2025), and is sufficient for paired bootstrap tests to detect the observed effect sizes (2–5 EM) at $p < 0.05$, as confirmed in Appendix L.

Baselines and Evaluation Protocol. We compare against methods spanning multiple RAG paradigms:

- **No retrieval**: Direct prompting (closed-book).
- **Standard RAG**: Naive RAG (top-k concatenated).
- **Adaptive RAG**: Self-RAG (Asai et al., 2024), IRCoT (Trivedi et al., 2023).
- **Prompt-based agentic**: Search-o1 (Li et al., 2025).
- **RL-based agentic**: Search-R1 (Jin et al., 2025), R1-Searcher (Song et al., 2025), ReARTeR (Sun et al., 2025), EviNote-RAG (Dai et al., 2025).

All methods share a canonical retrieval setup: E5-base-v2 (Wang et al., 2022) over the 2018 Wikipedia dump (Karpukhin et al., 2020). CORAL and prompt-based baselines use Qwen2.5-7B-Instruct as the base model. For Self-RAG and IRCoT, we follow their official prompting strategies, and Search-o1 is allowed the same 3-turn retrieval budget as the RL methods. For RL-based baselines, we evaluate their officially released checkpoints without further fine-tuning, as they were trained on

compatible data splits (NQ + HotpotQA). All four RL baselines (Search-R1, R1-Searcher, ReARTeR, EviNote-RAG) use Qwen2.5-7B-Instruct as their base model, the same E5-base-v2 retriever with top-3 passages, and a maximum of 3 retrieval turns, ensuring a fair comparison. To ensure robust evaluation, we run inference three times per baseline with different decoding seeds and report the median EM/F1, providing variance reduction symmetric to CORAL's 3-seed training average.

Implementation details. In our implementation, we retrieve the top-3 passages per query. The NLI model for $R_{\text{grounding}}$ is DeBERTa-v3-large (He et al., 2023) fine-tuned on MNLI. R_{format} is computed via regex matching against expected structural markers for each step. We set $\lambda_1 = 1.0$, $\lambda_2 = 0.3$, $\lambda_3 = 0.1$, and sample $G = 8$ rollouts per query during GRPO training. Training is conducted on 8 NVIDIA A100 GPUs using the verl framework (Sheng et al., 2024) for 800 steps with a learning rate of 1×10^{-6}, KL penalty coefficient of 0.001, and maximum sequence length of 4096 tokens.

Metrics. We report Exact Match (EM) and token-level F1 as answer quality metrics, following prior work (Jin et al., 2025; Dai et al., 2025). In addition, we conduct human evaluations of intermediate output quality in our Analysis (§4.4), measuring claim faithfulness and relational state annotation accuracy.

B Trajectory Format and Output Schemas

This section documents the system prompt and the output schemas that define CORAL's trajectory structure. During inference, the model receives *only* the system prompt (Section B.1). The stage-specific blocks in Sections B.2–B.4 are output schemas encouraged by R_{format} during training, not additional prompts injected at inference time. We present them here to specify the expected output format for reproducibility.

B.1 System Prompt (Agentic Loop)

The system prompt defines the overall iterative retrieve-organize-reason behavior:

```
You are a question-answering agent with
access to a search tool. To answer the
user's question, you will iteratively
search for evidence and organize it
before producing a final answer.

At each step, you may either:
(1) Issue a search query to gather more
evidence, OR
(2) Produce a final answer if sufficient
evidence has been collected.

When you retrieve documents, you MUST
process them through the following
three-stage evidence organization
pipeline before deciding your next
action:

Stage 1: DECOMPOSE - Extract atomic
claims from passages
Stage 2: ORGANIZE - Filter, merge, and
chain claims
Stage 3: NOTE - Annotate each claim
with its role

After organization, review your
accumulated notes. If critical
information is missing, formulate
a targeted follow-up search query.
Otherwise, synthesize the notes into a
final answer.

Format your output using the structural
tags below.
```

B.2 Stage 1: Claim Decomposition

```
<decompose>
Given the query and retrieved passages
below, decompose each passage into
atomic claims.

Requirements for each claim:
- ATOMIC: Express exactly one factual
proposition
- SELF-CONTAINED: Disambiguate all
pronouns, abbreviations, and implicit
references so the claim is interpretable
without surrounding context
- SOURCE-GROUNDED: Cite the source
document
- VERIFIABLE: Must be independently
checkable as true/false

Do NOT include opinions, rhetorical
statements, or meta-information about
the document itself.

Query: {query}
Passages:
[Doc1] {passage_1}
[Doc2] {passage_2}
[Doc3] {passage_3}

Output format:
[c1] <claim text> [Doc1]
[c2] <claim text> [Doc1]
[c3] <claim text> [Doc2]
...
</decompose>
```

Example. Given the passage *"Toni Morrison was awarded the Nobel Prize in Literature in 1993, becoming the first African-American woman to receive it."* from Doc3, with the query *"What university did the 1993 Nobel Literature laureate teach at?"*, the model produces:

> [c1] Toni Morrison was awarded the Nobel Prize in Literature in 1993. [Doc3]

[c2] Toni Morrison was the first African-American woman to receive the Nobel Prize in Literature. [Doc3]

Note that "She" is disambiguated to "Toni Morrison" and the compound sentence is split into two independent claims.

B.3 Stage 2: Claim Organization

```
<organize>
Given the query, the newly decomposed claims, and any existing notes from previous rounds, organize all evidence for answering.

If existing notes are provided, treat them as additional claims when discovering relationships. A new claim may OVERLAP with an existing note (merge to strengthen provenance) or BRIDGE with an existing note (extend a reasoning chain across rounds).

Step A - Relevance Triage:
Identify claims that are NON-CONTRIBUTORY to answering the query. A claim is non-contributory only if it has NEITHER:
(a) direct utility (contains information that directly answers the query), NOR
(b) indirect utility (serves as an intermediate step in a reasoning chain toward the answer).
Be conservative: when uncertain, RETAIN the claim.

Step B - Relationship Discovery (among retained claims only):
For each pair of retained claims, determine their query-conditioned relational state:

- OVERLAPPING: Both claims provide the same or redundant information for answering the query.
- BRIDGING: One claim introduces an entity or fact needed to interpret the other, forming a directed logical dependency for multi-hop reasoning. Specify the direction: ci -> cj means ci provides a premise for cj.
- ISOLATED: The claim is relevant but has no structural dependency with any other retained claim.

Step C - Structural Resolution:
Apply the following resolutions based on identified states:
- EXCLUDE: Remove non-contributory claims
- MERGE: Fuse overlapping claims into one representative claim with combined provenance
- CHAIN: Link bridging claims with directed edges, specifying the shared entity or logical link
- RETAIN: Keep isolated claims unchanged

Query: {query}
New claims: {claims from Stage 1}
Existing notes (from previous rounds, if any): {accumulated notes or "None"}

Output format:
[Excluded] c2, c5 (reason: <brief justification>)
[Merge] c1 + c4 -> C1: "<merged claim text>" [Doc1, Doc3]
[Chain] C1 -> c3 (via <bridge entity>)
[Isolated] c6
</organize>
```

B.4 Stage 3: Per-Claim Note Generation

```
<note>
For each retained/merged/chained claim from the organization stage, generate a usage note. Each note must specify:

(1) ROLE: One of the following:
  - Entry point: Starting claim of a reasoning chain
  - Bridge: Intermediate step connecting other claims
  - Answer-bearing: Directly contains the answer
  - Supporting: Provides confirming evidence
  - Independent: Standalone relevant fact

(2) SOURCE STRENGTH: Number of independent sources supporting this claim

(3) CONNECTIONS: What this claim follows from or leads to (if part of a chain)

(4) USAGE: How the downstream generator should use this claim when formulating the answer

Query: {query}
Organized claims: {output from Stage 2}

Output format:
C1: "<claim text>" [Doc1, Doc3]
  Note: Role=Entry point. Sources=2. Leads to c3 via <entity>. Use: identifies the target entity for the next reasoning step.

c3: "<claim text>" [Doc2]
  Note: Role=Answer-bearing. Sources=1. Follows from C1. Use: provides the final answer directly.

c6: "<claim text>" [Doc1]
  Note: Role=Independent. Sources=1. Use: supporting context, not required for the primary reasoning path.
</note>
```

B.5 Search Decision and Answer Generation

After generating notes, the model decides whether to search again or answer:

```
<decision>
Review the accumulated notes. Determine whether sufficient evidence exists to answer the query.

If a reasoning chain is INCOMPLETE (e.g., a bridging claim has no follow-up, or the answer-bearing claim is missing), formulate a targeted search query to
```

```
fill the gap:
<search> {new query targeting the missing information} </search>

If the evidence is SUFFICIENT, produce the final answer:
<answer> {concise answer} </answer>
</decision>
```

B.6 Complete Trajectory Example

Below is a condensed example of a complete trajectory for the query *"Where was the director of Parasite born?"*:

```
<search> director of Parasite film
</search>

[Retrieved: Doc1, Doc2, Doc3]

<decompose>
[c1] Bong Joon-ho directed the 2019 film
Parasite. [Doc1]
[c2] Parasite won the Palme d'Or at
Cannes 2019. [Doc1]
[c3] Bong Joon-ho directed the film
Parasite. [Doc2]
[c4] Parasite grossed over $250 million
worldwide. [Doc3]
</decompose>

<organize>
[Excluded] c2 (award info, not about
birthplace), c4 (box office, irrelevant)
[Merge] c1 + c3 -> C1: "Bong Joon-ho
directed the 2019 film Parasite." [Doc1,
Doc2]
[Isolated] (none)
</organize>

<note>
C1: "Bong Joon-ho directed the 2019 film
Parasite." [Doc1, Doc2]
  Note: Role=Entry point. Sources=2.
Use: identifies the director; need
birthplace next.
</note>

<search> Bong Joon-ho birthplace born
</search>

[Retrieved: Doc4, Doc5, Doc6]

<decompose>
[c5] Bong Joon-ho was born in Daegu,
South Korea. [Doc4]
[c6] Bong Joon-ho was born on September
14, 1969 in Daegu. [Doc5]
[c7] Bong Joon-ho studied sociology at
Yonsei University. [Doc5]
</decompose>

<organize>
[Excluded] c7 (education, not about
birthplace)
[Merge] c5 + c6 -> C2: "Bong Joon-ho was
born on September 14, 1969, in Daegu,
South Korea." [Doc4, Doc5]
[Chain] C1 -> C2 (via Bong Joon-ho)
</organize>

<note>
C1: "Bong Joon-ho directed the 2019 film
Parasite." [Doc1, Doc2]
  Note: Role=Entry point. Sources=2.
Leads to C2. Use: establishes director
identity.
C2: "Bong Joon-ho was born on September
14, 1969, in Daegu, South Korea." [Doc4,
Doc5]
  Note: Role=Answer-bearing. Sources=2.
Follows from C1. Use: provides
birthplace directly.
</note>

<answer> Daegu, South Korea </answer>
```

C Training Details

Table 5 lists the full set of hyperparameters used for GRPO training.

Hyperparameter	**Value**
Base model	Qwen2.5-7B-Instruct
Optimizer	AdamW
Learning rate	1×10^{-6}
LR schedule	Cosine with warmup
Warmup steps	30
Training steps	800
Batch size	128 queries
Rollouts per query (G)	8
Max sequence length	4096 tokens
Max retrieval turns	3
KL penalty coefficient	0.001
λ_1 (R_{answer})	1.0
λ_2 ($R_{\text{grounding}}$)	0.3
λ_3 (R_{format})	0.1
Temperature (sampling)	0.7
Top-p (sampling)	0.95
Retriever	E5-base-v2
Passages per query	3
NLI model ($R_{\text{grounding}}$)	DeBERTa-v3-large-MNLI
Hardware	8× NVIDIA A100 80GB
Framework	verl
Training time	∼30 hours

Table 5: Training hyperparameters for CORAL.

Training data. We train on the combined training sets of NQ and HotpotQA, totaling approximately 150K queries. During each training step, we sample a batch of 128 queries and generate $G = 8$ rollout trajectories per query. Each trajectory includes the full retrieve-organize-reason loop with up to 3 retrieval turns.

Reward implementation. R_{answer} is computed via exact match between the predicted and gold answers. $R_{\text{grounding}}$ computes the entailment probability from DeBERTa-v3-large fine-tuned on MNLI, averaged over all decomposed claims in the trajectory. R_{format} checks via regex whether the trajectory contains the expected structural markers for both the global agentic actions and the internal CORAL operations. Specifically, it verifies the presence of search query tags, claim decomposition tags, relational state predictions, per-claim notes,

and a final answer block. $R_{\text{format}} = 1$ if all markers are present, and 0 otherwise.

Inference. At inference time, we use greedy decoding with temperature 0. The retrieval setup remains identical to training: E5 retriever with top-3 passages and up to 3 retrieval turns.

Iterative evidence accumulation. Across multiple retrieval turns, CORAL maintains a cumulative note buffer. After each retrieval round, newly decomposed claims are organized jointly with the existing notes from previous rounds. This allows cross-round relationship discovery: a claim retrieved in round 2 may be identified as bridging with a claim from round 1. The model decides to issue a follow-up search when its notes indicate an incomplete reasoning chain (e.g., an entry-point claim with no answer-bearing follow-up). The maximum of 3 retrieval turns bounds context growth while allowing sufficient evidence gathering for 3–4 hop questions.

Format reward details. R_{format} verifies the presence of the following structural markers via regex: (1) at least one `<search>` tag; (2) `<decompose>` block with numbered claims and source citations; (3) `<organize>` block with at least one operation (Excluded/Merge/Chain/Isolated); (4) `<note>` block with Role annotations; and (5) `<answer>` tag. The reward is binary: $R_{\text{format}} = 1$ if all markers are present and well-formed, and 0 otherwise. This design ensures the model produces interpretable intermediate structures without constraining their semantic content.

D Per-dataset Ablation Breakdown

Table 6 reports the per-dataset ablation results that are averaged in Table 2.

E Variance Across Training Seeds

Table 7 reports the standard deviation of CORAL's EM scores across 3 training seeds for each dataset. All standard deviations are $\leq$0.6 EM, confirming stable training.

F Performance by Question Type

To understand where CORAL's gains originate, we categorize test questions by reasoning requirement using dataset-provided hop-count metadata and report per-category improvements over the strongest baseline. Single-hop questions come from NQ, TriviaQA, and PopQA; multi-hop questions from HotpotQA, 2Wiki, MuSiQue, and Bamboogle, further split by number of required reasoning steps. High-distractor questions are an orthogonal subset where the majority of retrieved passages are topically related but do not contain answer-relevant information, identified using gold supporting-fact annotations where available. Table 8 summarizes the results.

The gains scale with reasoning complexity: single-hop questions benefit marginally (+0.8 EM), while 2-hop and 3+ hop questions see progressively larger improvements (+3.8 and +5.2 EM, respectively). This trend confirms that the bridging chain mechanism becomes increasingly important as reasoning chains grow longer, explicitly constructing cross-document paths that passage-level methods cannot capture. High-distractor questions also benefit substantially (+2.8 EM), where relevance filtering effectively excludes misleading claims before they reach the generator.

G Efficiency Analysis

Table 9 compares the computational cost and accuracy of each method on HotpotQA. CORAL uses approximately 3,200 total trajectory tokens per query (including retrieved passages, intermediate structures, and generated output), resulting in higher latency than simpler baselines. However, the accuracy gains over EviNote-RAG (+4.0 EM) and Naive RAG (+20.0 EM) justify this additional computational cost. A cost-normalized comparison in Section P further demonstrates that simply scaling the compute budget of baselines to match CORAL does not yield comparable performance, confirming that the improvements stem from our explicit organizational structure rather than increased generation length.

Per-stage token breakdown. On HotpotQA, CORAL's average 3,200 trajectory tokens per query are distributed as follows: search queries and retrieved passages account for $\sim$900 tokens (28%), claim decomposition for $\sim$600 tokens (19%), organization (triage + relation + resolution) for $\sim$500 tokens (16%), notes for $\sim$400 tokens (12%), and the answer decision for $\sim$100 tokens (3%). The remaining $\sim$700 tokens (22%) arise from second- and third-round retrieval in multi-hop questions. The NLI model for $R_{\text{grounding}}$ adds negligible inference-time cost since it is only used during training to compute rewards; at inference, the model gener-

Setting	Multi-hop QA								Single-hop QA						Avg	
	HotpotQA		2Wiki		MuSiQue		Bamboogle		NQ		TriviaQA		PopQA			
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
CORAL (Full)	53.0	59.4	54.0	58.0	28.2	37.0	46.4	56.0	52.4	56.8	70.8	77.8	47.6	49.6	50.3	56.4
– Notes	48.8	54.6	48.4	52.8	24.2	32.2	41.6	50.0	50.4	54.8	68.4	74.6	47.2	49.2	47.0	52.6
– Organization	45.0	51.2	43.2	48.6	20.4	26.8	37.4	45.6	48.8	55.0	66.2	72.4	47.0	49.0	44.0	49.8
– Org – Notes	42.4	48.8	39.4	45.2	17.6	22.4	33.8	38.6	47.4	54.4	64.4	69.8	46.2	48.4	41.6	46.8
– $R_{\text{grounding}}$	51.2	57.0	52.0	55.8	26.4	34.6	44.2	53.4	51.4	55.6	69.4	76.0	45.8	47.2	48.6	54.2
Prompt-only (no RL)	24.8	28.6	23.4	27.2	8.2	14.4	22.4	29.0	30.6	35.4	38.2	41.8	28.8	32.2	25.2	29.8

Table 6: Per-dataset ablation results across all seven benchmarks.

Dataset	Mean EM	Std
HotpotQA	53.0	0.4
2WikiMultiHopQA	54.0	0.6
MuSiQue	28.2	0.5
Bamboogle	46.4	0.6
NQ	52.4	0.3
TriviaQA	70.8	0.2
PopQA	47.6	0.4

Table 7: Standard deviation of CORAL EM across 3 training seeds.

Question Type	Δ EM	Δ F1
Single-hop direct	+0.8	+1.2
Multi-hop 2-hop	+3.8	+4.6
Multi-hop 3+ hop	+5.2	+6.4
High distractor	+2.8	+3.4

Table 8: Performance improvement of CORAL over the best baseline, broken down by question type.

Method	Traj. Tokens	Latency	EM	F1
Naive RAG	500	1.2s	33.0	37.1
Search-R1	2,800	6.5s	42.0	46.4
EviNote-RAG	1,800	4.2s	49.0	55.7
CORAL	3,200	7.8s	53.0	59.4

Table 9: Efficiency comparison on HotpotQA.

Source Stage	%
Retrieval (missing from retrieved passages)	41
Claim organization	28
– Incorrect filtering (useful claim excluded)	12
– Missed bridging (chain not constructed)	11
– Wrong merge (non-equivalent claims fused)	5
Answer generation (wrong answer from correct notes)	22
Claim decomposition (unfaithful)	9

Table 10: Attribution of remaining errors by pipeline stage on HotpotQA (100 manually inspected incorrect predictions).

ates the full trajectory without NLI calls. The mean number of retrieval turns is 1.8 on single-hop datasets and 2.1 on multi-hop datasets; only 12% of multi-hop queries use all 3 turns.

H Remaining Error Attribution

To identify bottlenecks for future improvement, we manually inspect 100 incorrect predictions from CORAL on HotpotQA and trace each to the earliest pipeline stage where the error originates. Table 10 reports the results.

Retrieval accounts for 41% of errors, confirming that when the required evidence is absent from the retrieved passages, no downstream processing can compensate. Within the organization stage, incorrect filtering (12%) and missed bridging (11%) are the dominant failure modes, while wrong merges are relatively rare (5%). The filtering errors typically involve claims with indirect utility (e.g., intermediate steps in a reasoning chain) being mistakenly excluded, while missed bridging errors arise when the model fails to identify the shared entity connecting two claims across documents. Claim decomposition errors are rare (9%), validating the effectiveness of $R_{\text{grounding}}$. These results suggest that pairing CORAL with stronger retrievers and improving conservative bridging detection are the most promising directions for future gains.

I Model Size Scaling

To evaluate generalizability across model scales, we train and evaluate CORAL with Qwen2.5-3B-Instruct in addition to the default 7B model. Table 11 reports the results.

CORAL consistently outperforms both baselines at both scales, improving over EviNote-RAG by 2.8 EM at 3B and 2.4 EM at 7B. The gains transfer across model sizes, confirming that the claim-level organization system is not specific to a particular model capacity.

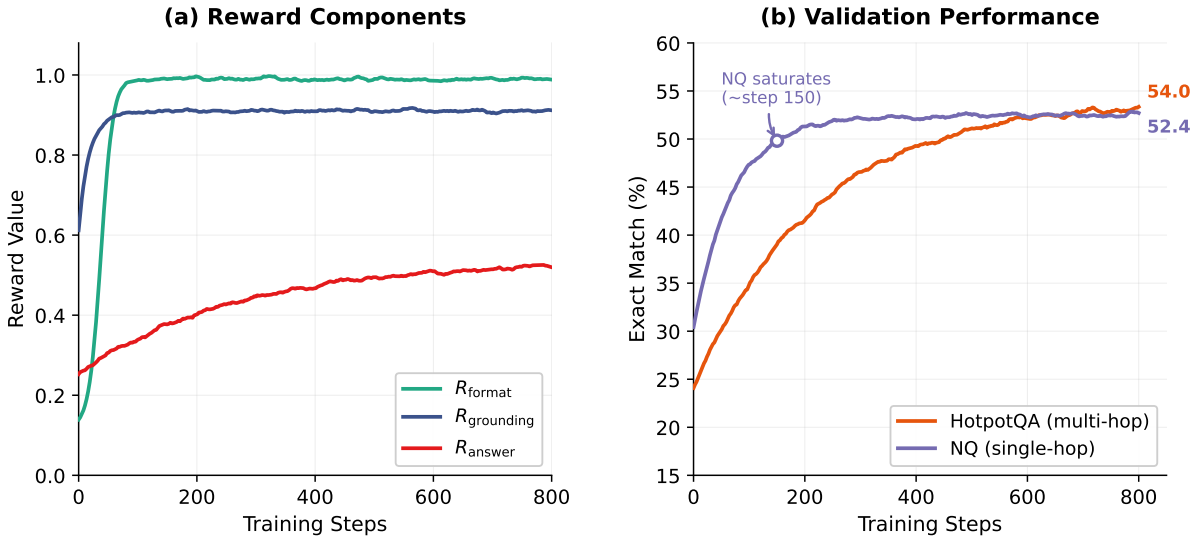

Figure 4: Training dynamics of CORAL. (a) Reward curves: $R_{\text{grounding}}$ converges within 50 steps, R_{format} reaches near-perfect compliance by step 80, while R_{answer} improves gradually throughout training. (b) Validation EM: NQ (single-hop) saturates early (∼step 150), while HotpotQA (multi-hop) improves steadily throughout, reflecting the progressive learning of evidence organization strategies.

Base Model	Method	Avg EM	Avg F1
Qwen2.5-3B	Naive RAG	24.6	27.4
	EviNote-RAG	38.4	43.6
	CORAL	41.2	46.6
Qwen2.5-7B	Naive RAG	31.0	33.2
	EviNote-RAG	47.9	53.9
	CORAL	50.3	56.4

Table 11: Scaling analysis. CORAL consistently outperforms baselines at both 3B and 7B scales. The 7B results are taken from Table 1.

J Training Dynamics

We observe distinct learning phases during GRPO training (Figure 4). The grounding reward ($R_{\text{grounding}}$) converges within the first 50 steps, indicating that faithful claim decomposition is relatively easy to learn. The format reward (R_{format}) reaches near-perfect compliance by step 80, after which the model consistently produces well-structured outputs. The answer reward (R_{answer}) improves more gradually throughout training, reflecting the model's progressive discovery of effective organization strategies such as relevance filtering, overlapping merge, bridging chain construction, and isolated claim retention.

Validation EM on HotpotQA shows steady improvement throughout training, while NQ saturates earlier (around step 150). This is consistent with the intuition that multi-hop reasoning benefits more from extended optimization of evidence organization, whereas single-hop questions are largely solved once basic claim decomposition and filtering are learned.

K Qualitative Case Studies

We provide two qualitative case studies to illustrate how CORAL's claim-level organization differs from passage-level RAG methods. The first case study focuses on multi-hop reasoning with a same-nationality distractor, where explicit filtering and bridging are essential. The second case study focuses on a single-hop question with redundant evidence, where overlapping merge and isolated claim retention are most relevant.

K.1 Main Case Study: Multi-hop Reasoning with Misleading Distractors

For the query *"In what city was the director of the film that won the 2020 Academy Award for Best Picture born?"* (Figure 5), retrieval returns passages about *Parasite*, Bong Joon-ho, and Park Chan-wook. The correct reasoning path requires first identifying *Parasite* as the 2020 Best Picture winner, then linking *Parasite* to its director Bong Joon-ho, and finally extracting Bong Joon-ho's birthplace. However, the retrieved passages also contain misleading facts: Bong Joon-ho's childhood relocation and education in Seoul, and a separate passage about Park Chan-wook, another South Korean director born in Seoul.

- **Naive RAG**: Concatenating the retrieved passages exposes the model to repeated mentions of Seoul. Although one passage states that Bong Joon-ho was born in Daegu, South Korea, the model overweights the more salient Seoul-related context and answers "Seoul."

- **EviNote-RAG**: Its passage-level summary correctly identifies Bong Joon-ho as the director of *Parasite*, but the free-form summarization step collapses Bong's biography into Seoul-related facts and drops the Daegu birthplace fact during compression. The generator, conditioned only on the summary, answers Seoul.

- **CORAL**: It decomposes the passages into atomic claims, separates birthplace evidence from related biographical facts, and excludes claims about the wrong relation (c4, c5) or wrong director (c6, c7). It explicitly chains "*Parasite* won Best Picture" (c1) to "*Parasite* was directed by Bong Joon-ho" (c2), and then to "Bong Joon-ho was born in Daegu, South Korea" (c3), which it marks as answer-bearing. This structured organization allows the model to return "Daegu, South Korea."

This example illustrates a common failure pattern in passage-level methods: facts that are topically related to the target entity can be mistaken for answer evidence when the required predicate is not explicitly checked. Here, Seoul is relevant to Bong Joon-ho's biography but not to his birthplace, while Park Chan-wook matches the broad type "South Korean director" but is not the director of *Parasite*. Claim-level organization prevents these errors by isolating each factual predicate as a separate claim and verifying whether it satisfies the target relation before passing it to the generator.

K.2 Supplementary Case Study: Overlapping Merge and Isolated Retention

The second case study (Figure 6) shows a single-hop query where multiple retrieved passages restate the same answer-bearing fact. For the query *"When was Marie Curie awarded her first Nobel Prize?"*, the retrieved passages include several claims about Marie Curie's 1903 Nobel Prize, as well as additional facts about her later 1911 Chemistry Nobel Prize.

CORAL first decomposes the retrieved passages into atomic claims. It then merges overlapping claims (c1, c3, c4) that all support the answer "1903," excludes claims about the unrelated 1911 Chemistry Nobel Prize (c5, c6), and retains an isolated supporting claim (c2) that provides useful but non-essential context. This example shows that CORAL's organization module is not limited to multi-hop bridging: it also improves single-hop evidence processing by reducing redundancy and preserving partially relevant context.

L Statistical Significance Tests

Table 12 reports paired bootstrap test results (10K resamples) comparing CORAL against EviNote-RAG on each dataset.

Dataset	Δ EM	Δ F1	p-value	Sig.
HotpotQA	+4.0	+3.7	<0.001	**
2WikiMultiHopQA	+4.6	+4.4	<0.001	**
MuSiQue	+4.2	+3.4	<0.001	**
Bamboogle	+4.0	+3.2	0.003	**
NQ	0.0	+0.5	0.18	n.s.
TriviaQA	+0.5	+1.3	0.04	*
PopQA	−0.4	+0.5	0.42	n.s.

Table 12: Per-dataset paired bootstrap significance tests (CORAL vs. EviNote-RAG, 10K resamples). **: $p < 0.01$; *: $p < 0.05$; n.s.: not significant. p-values are computed on EM; for NQ, where $\Delta\text{EM} = 0$, we report the F1 p-value instead.

The improvements are highly significant on all multi-hop benchmarks, confirming that CORAL's structured claim organization provides reliable gains when multi-step reasoning is required. On single-hop benchmarks, the differences are small and mostly not significant, indicating that CORAL neither substantially helps nor hurts when evidence organization is straightforward.

M Relational State Evaluation Methodology

We describe the full evaluation protocol for relational state prediction quality (Table 4 in the main text).

Precision evaluation. We randomly sample 50 model predictions per relation type (overlapping, bridging, irrelevant; 150 total) from the HotpotQA test set outputs. For each sampled prediction, three annotators independently judge whether the predicted relational state is correct given the query and the two claims involved. Annotators are shown the original source passages for reference. Majority vote determines the gold label.

Recall evaluation. We draw a separate set of 80 queries from the HotpotQA test set. For each query, annotators identify all claims participating in at least one non-isolated relation and annotate all pairs among these candidates, yielding 320 annotated claim pairs (average 4 per query). We then measure how many ground-truth relations were correctly detected by the model.

Independence of samples. The precision and recall samples are drawn from different queries to prevent data contamination. This ensures that precision estimates ("when the model predicts a relation, is it correct?") and recall estimates ("of all true relations, how many does the model find?") are statistically independent.

Sample size justification. The 320 pairs used for recall annotation yield 95% confidence intervals of approximately ±4–5 percentage points at the observed recall levels (72–85%). This precision is sufficient to distinguish the observed differences across relation types (e.g., overlapping recall of 84.6% vs. bridging recall of 72.8%, a gap of 11.8 points that exceeds the confidence interval width). For precision, 50 samples per type yield ±7–10 percentage point confidence intervals. While this wider interval limits strict statistical comparisons between types, it provides a sufficiently tight bound to confirm that the absolute precision of all predicted relations remains high.

N Organization Quality Under Outcome-Driven RL

Since no reward component directly supervises organization correctness, we verify empirically that the outcome-driven reward signal induces progressively better organization during training.

Evidence 1: Organization quality improves with training. We checkpoint the model at steps 0, 100, 200, 400, and 800, and measure relational state prediction accuracy on a fixed set of 100 HotpotQA validation examples with human-annotated ground-truth relations. Table 13 shows that organization quality improves monotonically alongside answer accuracy on the same 100 examples, despite no reward directly targeting organization correctness.

Evidence 2: Organization errors correlate with answer failures. We analyze 200 trajectories from the final model on HotpotQA validation (100 correct, 100 incorrect answers) and manually annotate whether the organization stage contains errors. Among correct-answer trajectories, 91% have fully correct organization. Among incorrect-answer trajectories, 67% contain at least one organization error (wrong merge, missed bridge, or incorrect exclusion). This strong correlation confirms that R_{answer} provides a reliable gradient signal for organization quality: faulty organization almost always produces a wrong answer and is penalized.

Step	Relation F1	Triage Acc.	EM
0 (init)	31.2	52.4	23.0
100	58.6	71.8	37.0
200	68.4	78.2	45.0
400	74.8	83.6	50.0
800 (final)	79.2	86.4	54.0

Table 13: Organization quality and answer accuracy across training steps. Relation F1 measures correctness of overlapping/bridging predictions; Triage Acc. measures precision of irrelevant filtering. Both improve with training despite receiving no direct reward.

Evidence 3: Counterfactual intervention on organization. To test whether learned organization is load-bearing rather than decorative, we take the final trained model's outputs on 100 HotpotQA examples and manually perturb the organization stage while keeping all other components intact:

- **Shuffle relations**: Randomly reassign overlapping/bridging/isolated labels among retained claims. EM drops from 54.0 to 35.0 (−19.0).
- **Reverse chains**: Flip all bridging directions. EM drops to 39.0 (−15.0).
- **Remove exclusions**: Restore all excluded claims. EM drops to 45.0 (−9.0).
- **Disable merge**: Keep overlapping claims as separate entries. EM drops to 50.0 (−4.0).

These large performance drops under targeted perturbation confirm that the model's organization decisions are semantically meaningful and causally contribute to answer correctness. The magnitude of each drop also reveals a hierarchy: relational state assignment and chain direction matter most, followed by relevance filtering, then deduplication via merge.

Summary. These three analyses are mutually reinforcing: organization quality improves monotonically with training (Evidence 1), incorrect organization correlates strongly with answer failures (Evidence 2), and perturbing the learned organization causes large accuracy drops (Evidence 3). Together,

they indicate that the outcome-driven reward provides an effective learning signal for organization quality through the deterministic information flow from organization to notes to final answer.

O Bridging Error Impact Analysis

The relational state evaluation (Table 4) shows that bridging predictions have the lowest F1 (77.1), with errors including direction reversal and missed intermediate steps. To understand how these errors affect downstream answer quality, we analyze the 50 precision-sampled bridging predictions and cross-reference them with final answer correctness.

Error breakdown. Among the 9 incorrect bridging predictions (18% error rate out of 50 samples):

- **Direction reversal** (5 cases): The model correctly identifies that two claims are bridging-related but assigns the wrong direction (e.g., predicts $c_i \rightarrow c_j$ when the logical dependency is $c_j \rightarrow c_i$).
- **Missing intermediate** (3 cases): The model links two claims that require an intermediate step, skipping a necessary bridge entity.
- **Spurious bridge** (1 case): The model predicts a bridging relation between claims that are actually isolated.

Impact on final answers. Of the 9 bridging errors, 6 (67%) lead to incorrect final answers. Direction reversals are particularly harmful: 4 out of 5 reversed chains produce wrong answers because the reasoning path starts from the wrong premise. Missing intermediate errors lead to wrong answers in 2 out of 3 cases, as the incomplete chain fails to connect to the answer-bearing claim. The single spurious bridge does not harm the final answer because the incorrectly linked claim happens to be compatible with the correct reasoning path.

Implications. These results confirm that bridging chain quality is the most critical organization component for multi-hop QA, consistent with the intervention experiments in Appendix N where reversing chains causes the second-largest EM drop (-15.0). Improving bridging detection, particularly direction assignment, is the highest-priority target for future work.

P Cost-Normalized Comparison

A natural question is whether CORAL's gains simply reflect its higher token budget (3,200 tokens vs. 1,800 for EviNote-RAG). To test this, we evaluate EviNote-RAG and Search-R1 with an increased retrieval budget of 5 turns (vs. 3 for CORAL), allowing them to generate comparable total tokens. Table 14 reports the results on HotpotQA.

Method	Traj. Tokens	EM	F1
Search-R1 (3 turns)	2,800	42.0	46.4
Search-R1 (5 turns)	4,200	43.4	47.8
EviNote-RAG (3 turns)	1,800	49.0	55.7
EviNote-RAG (5 turns)	3,100	50.2	56.4
CORAL (3 turns)	3,200	53.0	59.4

Table 14: Cost-normalized comparison on HotpotQA. Baselines are given additional retrieval turns to match or exceed CORAL's token budget.

With 5 retrieval turns and comparable token budgets, Search-R1 gains only +1.4 EM and EviNote-RAG gains +1.2 EM. CORAL with 3 turns still outperforms both extended baselines by a substantial margin (+9.6 over Search-R1 5-turn, +2.8 over EviNote-RAG 5-turn). This confirms that CORAL's gains stem from structured evidence organization rather than simply consuming more tokens or retrieving more passages.

Q NLI Reward Correlation with Human Faithfulness

Since $R_{\text{grounding}}$ uses DeBERTa-v3-large fine-tuned on MNLI, a natural concern is whether this NLI model's entailment judgments align with human faithfulness assessments in our domain (Wikipedia factual claims). We measure the correlation between the NLI model's entailment probability and human binary faithfulness labels on the 600 annotated claims from Table 3.

Results. The NLI model achieves 89.3% agreement with human majority-vote labels (binary threshold at entailment probability > 0.5). Specifically:

- On claims labeled faithful by humans: NLI predicts entailment for 93.1% (true positive rate).
- On claims labeled hallucinated by humans: NLI predicts non-entailment for 78.4% (true negative rate).
- Point-biserial correlation between NLI probability and human label: $r = 0.74$ ($p < 0.001$).

The lower true negative rate (78.4%) indicates that the NLI model occasionally assigns high entailment to hallucinated claims, particularly when the claim is semantically plausible and partially

overlaps with the source. However, the overall correlation ($r = 0.74$) is strong enough to provide a meaningful training signal, and the human evaluation in Table 3 confirms that the model trained with this reward achieves 94.2% human-judged faithfulness.

R Claim Count Analysis

Since $R_{\text{grounding}}$ averages entailment probability over generated claims, a conservative strategy of producing fewer claims could trivially inflate this reward component. We verify that this reward hacking does not occur by tracking claim generation statistics across training and at convergence.

Claim count during training. Table 15 reports the average number of claims generated per query at different training checkpoints, measured on the same 200 HotpotQA validation examples.

Step	Avg Claims	R_{ground}	EM
0 (init)	4.2	0.61	25.5
100	6.8	0.78	39.0
200	7.4	0.84	45.5
400	7.8	0.89	50.0
800 (final)	7.6	0.91	53.5

Table 15: Average claims per query across training steps. Claim count increases during training and stabilizes, showing no evidence of reward hacking via claim reduction.

The model generates *more* claims as training progresses (4.2 → 7.6), not fewer. This trend reflects the model learning to extract finer-grained atomic facts from passages rather than collapsing toward conservative outputs. The slight decrease from step 400 to 800 (7.8 → 7.6) reflects improved relevance triage rather than grounding exploitation: the model learns to exclude genuinely irrelevant claims earlier.

Final model claim statistics. At convergence, the model generates an average of 7.6 claims per query on multi-hop benchmarks and 5.2 claims per query on single-hop benchmarks. The difference is expected: multi-hop questions require more evidence pieces. For comparison, manual inspection of 50 HotpotQA examples identifies an average of 8.1 ground-truth relevant claims per question in the retrieved passages, indicating that the model captures approximately 94% of relevant factual content.

Why $\mathbf{R}_{\text{answer}}$ prevents coverage hacking. The dominant R_{answer} ($\lambda_1 = 1.0$, vs. $\lambda_2 = 0.3$ for grounding) creates strong implicit pressure to maintain claim coverage. On multi-hop questions, omitting even one bridging claim breaks the reasoning chain and yields $R_{\text{answer}} = 0$, a penalty of 1.0 that far exceeds any grounding gain from having one fewer claim in the denominator (at most $\sim$0.04 with 7–8 claims). This asymmetry ensures that the model cannot profitably trade coverage for grounding score.

S Human Annotation Details

Two human evaluation tasks are reported in this paper: claim faithfulness evaluation (§4.4; 600 claims) and relational state prediction quality (§4.4; 150 precision samples + 320 recall pairs). Both tasks employed the same pool of three annotators.

Recruitment and qualifications. Annotators were recruited from graduate students with backgrounds in NLP or computational linguistics. All annotators had prior experience with annotation tasks involving natural language inference and information extraction.

Compensation. Annotators were compensated at an hourly rate above the local minimum wage. Total annotation time was approximately 25 hours across all tasks.

Consent and data use. Annotators were informed of the research purpose and the intended use of their annotations for aggregate evaluation metrics prior to participating. They were informed of their right to withdraw at any time without penalty. No personally identifying information about annotators is collected or released in this work.

Question: In what city was the director of the film that won the 2020 Academy Award for Best Picture born?

Naive RAG

<information>
Doc 1 (Title: "Parasite (2019 film)"): Parasite is a 2019 South Korean black comedy thriller film directed by Bong Joon-ho. The film won the Palme d'Or at the 2019 Cannes Film Festival and the Academy Award for Best Picture at the 92nd Academy Awards in February 2020, becoming the first non-English-language film to win Best Picture ...
Doc 2 (Title: "Bong Joon-ho"): Bong Joon-ho (born September 14, 1969) is a South Korean filmmaker, born in **Daegu, South Korea**. While Bong was in elementary school, the family relocated to Seoul ... He enrolled in Yonsei University in Seoul ... and is best known for Memories of Murder (2003), The Host (2006), Snowpiercer (2013), and Parasite (2019) ...
Doc 3 (Title: "Park Chan-wook"): Park Chan-wook (born 23 August 1963) is a South Korean filmmaker and former film critic, born in Seoul, South Korea. He is widely regarded as a leading figure in 21st-century world cinema, known for Joint Security Area (2000), Oldboy (2003), The Handmaiden (2016), and Decision to Leave (2022) ...
</information>
<think>
The question asks for the birth city of the director of the film that won Best Picture in 2020. The retrieved passages say that Parasite won the Academy Award for Best Picture, and that Parasite was directed by Bong Joon-ho. The passages about Korean directors mention both Daegu and Seoul, but Seoul appears repeatedly in the retrieved context and is strongly associated with the director's background and Korean film context. Therefore, the director was likely born in Seoul.
</think>
<answer>
Seoul
</answer>

Frequency bias → wrong answer

EviNote-RAG

<information>
[same three docs as Naive RAG]
</information>
<summary>
The 2020 Academy Award for Best Picture was won by *Parasite*, directed by *Bong Joon-ho*, a South Korean filmmaker. Bong Joon-ho's family relocated to Seoul during his childhood, and he later attended Yonsei University in Seoul, where he established his filmmaking career. *The central biographical evidence centers on his Seoul-based upbringing and education*.
</summary>
<answer>
Seoul
</answer>

Wrong predicate marked as key

CORAL

<information>
[same three docs as Naive RAG]
</information>
<decompose>
c1: Parasite won the 2020 Academy Award for Best Picture. [Doc 1]
c2: Parasite was directed by Bong Joon-ho. [Doc 1]
c3: Bong Joon-ho was born in Daegu, South Korea. [Doc 2]
c4: Bong Joon-ho's family relocated to Seoul during his elementary school years. [Doc 2]
c5: Bong Joon-ho attended Yonsei University in Seoul. [Doc 2]
c6: Park Chan-wook is a South Korean filmmaker. [Doc 3]
c7: Park Chan-wook was born in Seoul, South Korea. [Doc 3]
</decompose>

Claim decomposition

<organize>
[Triage]
c1, c2, c3 — on the target chain ✓
c4, c5 — about Bong's relocation/education, predicate ≠ born_in ✗
c6, c7 — about Park Chan-wook, ≠ director of target film ✗
[Exclude] c4, c5 — predicate mismatch ("relocated_to" / "attended" are not "born_in").
[Exclude] c6, c7 — bridging entity "Park Chan-wook" fails the target relation (not director of Parasite).
[Chain] c1 → c2 → c3 via entities "Parasite" → "Bong Joon-ho".
</organize>

Filter non-contributory claims

Construct bridging chains

<note>
c1 [Entry point]: Identifies the 2020 Best Picture film.
c2 [Bridge]: Identifies the director of that film.
c3 [Answer-bearing]: Provides the birth city of the target director.
</note>

Per-claim note generation

<answer>
Daegu, South Korea
</answer>

Correct answer

Figure 5: Case study on multi-hop reasoning with misleading distractors. While passage-level baselines are misled by spurious relations, CORAL explicitly filters distractor claims and constructs the correct reasoning chain.

Question: When was Marie Curie awarded her first Nobel Prize?

CORAL

<information>

Doc 1 (Title: "Nobel Prize in Physics"): Marie Curie was awarded the Nobel Prize in Physics in **1903**, shared with Pierre Curie and Henri Becquerel, for their work on radioactivity...

Doc 2 (Title: "Marie Curie"): In **1903**, the Nobel Prize in Physics was awarded to Marie Curie in recognition of her research on radiation phenomena...

Doc 3 (Title: "Marie Curie"): Marie Curie later won the Nobel Prize in Chemistry in 1911, making her the first person to receive Nobel Prizes in two different sciences...

</information>

<decompose>

c1: Marie Curie was awarded the Nobel Prize in Physics in 1903. [Doc 1]

c2: The 1903 Nobel Prize in Physics was shared with Pierre Curie and Henri Becquerel. [Doc 1]

c3: The 1903 Nobel Prize in Physics was awarded for research on radioactivity. [Doc 1]

c4: Marie Curie won the Nobel Prize in Physics in 1903 for her research on radiation. [Doc 2]

c5: Marie Curie won the Nobel Prize in Chemistry in 1911. [Doc 3]

c6: Marie Curie was the first person to receive Nobel Prizes in two different sciences. [Doc 3]

Claim decomposition

</decompose>

<organize>

[Triage]

c1, c2, c3, c4 — about the 1903 Physics Nobel ✓

c5, c6 — about the 1911 Chemistry Nobel ✗

Filter non-contributory claims

[Exclude] c5, c6 — concern the second Nobel Prize, not the first; non-contributory to the query.

[Merge] c1, c3, c4 → C1 — overlapping: Marie Curie won the Nobel Prize in Physics in 1903 for her research on radiation [Doc 1, Doc 2].

[Isolated] c2 — isolated: names co-recipients.

Retain isolated claims

Merge overlapping claims

</organize>

<note>

C1 [Answer-bearing]: Directly provides the year and field of the first Nobel Prize.

c2 [Supporting context]: Names co-recipients.

Per-claim note generation

</note>

<answer>

1903

</answer>

Correct answer

Figure 6: Case study on single-hop evidence organization. CORAL merges overlapping claims to reduce redundancy and excludes unrelated facts, while retaining isolated supporting context.