DIRECTION-AWARE ITERATIVE GRAPH RETRIEVAL VIA ONLINE CONVEX OPTIMIZATION FOR LONG-DOCUMENT MULTI-HOP QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-document multi-hop question answering requires retrieving evidence whose relevance may become apparent only after intermediate reasoning steps. Graph-based retrieval can connect such evidence, but diffusion from a fixed seed distribution may assign limited probability mass to distant evidence or disperse it across competing outgoing edges. We propose RCT-PPR, an iterative retrieval framework that adapts both the restart distribution and local transition probabilities of Personalized PageRank. An LLM controller identifies new seed entities and a semantic target direction from the retrieved context. The target induces a linear alignment loss over each node's outgoing probability simplex, enabling a multiplicative-weights update that preserves normalized transitions and accumulates semantic feedback across iterations. We characterize the cumulative alignment loss of this update through a local static-regret bound against the best fixed outgoing distribution in hindsight. On three multi-hop QA datasets from LongBench, RCT-PPR achieves 53.3 average F1, exceeding HippoRAG 2 by 6.4 points. Ablations reduce average F1 to 48.1 with a single retrieval round and to 51.3 without the OCO-based update, supporting the contributions of iterative retrieval and transition adaptation.

1 INTRODUCTION

Multi-hop question answering over long documents requires assembling evidence distributed across multiple text chunks. The initial question may not identify all entities or relations needed to complete the evidence chain, so a retriever must respond to information discovered during reasoning. For example, the question *"Where was the mother of the founder of SpaceX born?"* can be decomposed into identifying the founder, his mother, and her birthplace. Figure 1 illustrates how these facts can occur in separate chunks and how updating the retrieval focus can make a previously distant fact accessible.

Graph-augmented retrieval methods, including HippoRAG and HippoRAG 2 (Gutiérrez et al., 2024; 2025), use extracted knowledge to propagate relevance beyond directly matched passages. Personalized PageRank (PPR) (Brin & Page, 1998) provides a mechanism for such propagation, but its behavior depends on both the restart distribution and the transition matrix. Contributions from longer walks are discounted, while branching can distribute probability mass among many competing neighbors. Consequently, a graph may contain the necessary evidence without ranking it highly enough for the answer generator.

Iterative retrieval offers a complementary mechanism. IRCoT (Trivedi et al., 2023) interleaves retrieval with intermediate reasoning, while other approaches refine retrieval through decomposition, aggregation, or knowledge triples (Press et al., 2023; Chu et al., 2024; Fang et al., 2025). These methods show the value of adapting retrieval to evolving information needs. Our focus is on how such reasoning feedback can update a graph diffusion operator: changing the restart seeds determines *where* retrieval is anchored, whereas changing local transition probabilities determines *how* relevance is routed from those anchors.

We introduce **RCT-PPR** (Reasoning-Controlled Transition Personalized PageRank), which combines these two forms of adaptation in an iterative graph retrieval loop. An LLM controller inspects

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

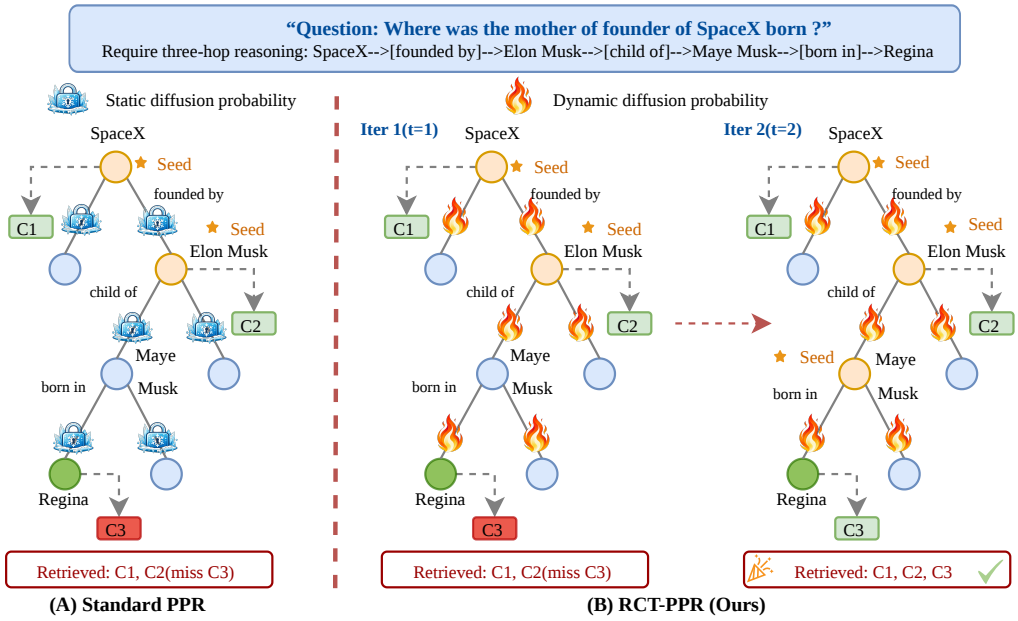

Figure 1: **Illustration of iterative graph retrieval.** (A) With a fixed seed set, PPR may rank nearby chunks above the distant evidence chunk C3. (B) After inspecting the initial context, the controller identifies an intermediate entity and updates the seeds and retrieval direction. A subsequent retrieval round can then reach C3.

retrieved evidence and proposes the next seed entities and a semantic target direction. For each node, similarity between that direction and its outgoing relation descriptions defines a linear loss over the outgoing probability simplex. A multiplicative-weights update (MWU) then adjusts the transition distribution while retaining information from earlier updates. The resulting entity scores are mapped to chunks and combined with dense retrieval scores.

The OCO formulation specifies the local objective and the role of historical feedback. Under bounded losses and fixed outgoing edge sets, the update has a static-regret bound of $\mathcal{O}(\sqrt{T \log d_{\max}})$ per node, where T is the feedback horizon and $d_{\max}$ is the maximum out-degree. This is a guarantee about cumulative semantic alignment with the observed controller targets. It neither assumes that those targets are correct nor guarantees discovery of an optimal reasoning path. We evaluate the practical effect of the update separately from its theoretical interpretation.

Contributions.

- We introduce an iterative graph retrieval framework that jointly adapts restart seeds and local transition probabilities using reasoning feedback.
- We formulate transition adaptation as per-node online linear optimization, derive its normalized MWU rule, and state a local static-regret guarantee with an explicit scope and horizon-dependent temperature choice.
- On three LongBench multi-hop QA datasets, RCT-PPR achieves 53.3 average F1 versus 46.9 for HippoRAG 2. Component and iteration ablations distinguish the benefit of the full retrieval loop from the additional contribution of the OCO-based update.

2 Related Work

Text and hierarchical retrieval. Sparse and dense retrievers, including BM25, DPR, Contriever, and GTR (Robertson & Zaragoza, 2009; Karpukhin et al., 2020; Izacard et al., 2021; Ni et al., 2022), rank passages by lexical or learned semantic relevance. RAPTOR (Sarthi et al., 2024) adds hierarchical summaries to expose information at different granularities. These retrieval mechanisms can

also be used within iterative systems. Our contribution concerns the adaptation of graph transitions during reasoning, rather than a replacement for text encoders or similarity search.

Graph-augmented and iterative retrieval. GraphRAG and LightRAG (Edge et al., 2024; Guo et al., 2024) use document structure to support retrieval and generation. The HippoRAG family (Gutiérrez et al., 2024; 2025) uses PPR for associative retrieval over extracted knowledge. Meanwhile, IRCoT, BeamAggR, and Self-RAG (Trivedi et al., 2023; Chu et al., 2024; Asai et al., 2023) integrate reasoning, aggregation, or reflection with retrieval. Particularly relevant is KiRAG (Fang et al., 2025), which performs knowledge-driven iterative retrieval using extracted triples. Thus, neither iteration nor the use of structured knowledge alone distinguishes our method. RCT-PPR specifically maintains and updates local PPR transition distributions across reasoning rounds, in addition to revising the restart seeds.

Online optimization. Online convex optimization (Hazan, 2016), Hedge (Freund & Schapire, 1997), and mirror descent (Nemirovskij & Yudin, 1983) provide established tools for sequential decision-making, including routing applications (Awerbuch & Kleinberg, 2004). We apply the standard entropic mirror-descent update to outgoing graph transitions, using controller-derived semantic alignment as the feedback signal. The novelty lies in this retrieval formulation and its integration with iterative reasoning; the underlying MWU rule and static-regret analysis are established online-learning tools.

3 Approach

We first distinguish the stored knowledge graph from the graph used for diffusion, then describe the retrieval loop and its local transition update.

3.1 Preliminaries

Graph representation. Given a document collection $\mathcal{D}$, we extract entity nodes $\mathcal{V}_{ent}$, chunk nodes $\mathcal{V}_{chk}$, directed relation edges, and entity-to-chunk associations. The stored representation is heterogeneous. In the default configuration, however, PPR operates on the *directed entity-relation subgraph* $G_{ent} = (\mathcal{V}_{ent}, \mathcal{E}_{rel})$; chunk associations are used to map entity scores back to text. Section 4.2.2 compares this configuration with diffusion graphs that also contain chunk nodes. Each relation edge has a normalized embedding obtained from its textual description.

Personalized PageRank. Let $\mathbf{s}$ be a normalized restart distribution over entity nodes, $\alpha \in (0, 1)$ the restart probability, and $\mathbf{W}$ a column-stochastic transition matrix, with W_{vu} denoting the probability of moving from u to v. PPR satisfies

$$\boldsymbol{\pi} = (1 - \alpha)\mathbf{W}\boldsymbol{\pi} + \alpha\mathbf{s}. \tag{1}$$

Its expansion, $\boldsymbol{\pi} = \alpha \sum_{k=0}^{\infty} (1 - \alpha)^k \mathbf{W}^k \mathbf{s}$, shows that contributions from length-k walks are discounted by $(1-\alpha)^k$. The probability at a particular node also depends on transition weights and the number of walks reaching it; hop distance alone does not determine its score. Updating $\mathbf{s}$ can shorten the effective distance to newly identified evidence, while updating $\mathbf{W}$ can redirect probability mass among competing outgoing edges.

3.2 Overall Framework

Offline construction and initialization. Documents are partitioned into overlapping chunks. An LLM extracts relational triples, and an encoder maps chunk text and relation descriptions to vectors. At query time, dense retrieval selects candidate triples, which are filtered by an LLM to initialize the seed distribution. Graph construction is performed offline and is separate from the query-time reasoning loop.

Retrieval and controller feedback. At round t, PPR uses the current seed distribution $\mathbf{s}_t$ and transition matrix $\mathbf{W}^{(t)}$. Retrieved chunks are added to a context memory $\mathcal{M}$. The controller reads this memory and the original question, estimates whether sufficient evidence has been collected,

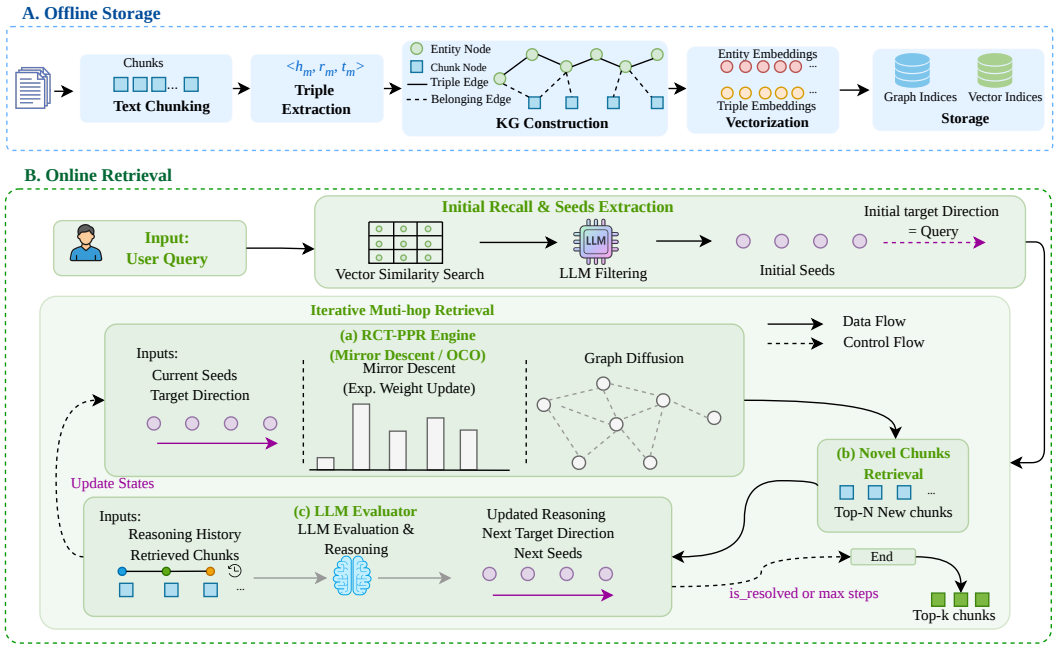

Figure 2: **Architecture of RCT-PPR.** (A) Offline processing extracts relational triples and entity-to-chunk associations, and stores graph and vector representations. (B) Online retrieval initializes seeds through triple retrieval and LLM filtering. Repeated PPR retrieval and controller feedback update the evidence memory, restart seeds, and local transition distributions. Entity-to-chunk aggregation and dense-score fusion produce the final top-K chunks.

and, if needed, proposes a new seed distribution and a textual target for the next round. The target embedding supplies the feedback for the transition update in Section 3.3. Retrieval stops when the controller returns `is_resolved` or the maximum iteration budget is reached. This stopping decision is a controller prediction, not a certificate that the evidence is complete.

Chunk ranking. Entity relevance is aggregated into chunk scores using IDF weighting and combined with dense semantic scores through a fusion weight β. Final selection is made from the accumulated candidate memory. Separating entity diffusion from chunk selection allows the retrieval process to use relation structure while retaining a text-based relevance signal.

3.3 Local Transition Adaptation

Local objective and feedback order. For a node v with $d_v \geq 1$ outgoing edges, let $\mathbf{w}_t^{(v)} \in \Delta^{d_v}$ be its transition distribution, where $\Delta^{d_v} = \{\mathbf{w} \geq 0 : \sum_j w_j = 1\}$. The graph and outgoing edge sets remain fixed during a query. The controller target produced after retrieval round t defines the loss vector

$$\ell_{t,j}^{(v)} = -\langle \mathbf{e}_{v,j}, \mathbf{r}_t \rangle, \qquad L_t^{(v)}(\mathbf{w}) = \langle \mathbf{w}, \boldsymbol{\ell}_t^{(v)} \rangle, \tag{2}$$

where the edge and target embeddings have unit Euclidean norm, so $\ell_{t,j}^{(v)} \in [-1, 1]$. This objective is linear and hence convex in $\mathbf{w}$. The distribution used at round t precedes the feedback $\mathbf{r}_t$; the updated distribution is used in the next round. Targets may depend on previously retrieved evidence, and need not be independent of the retrieval history.

Entropic mirror descent. For learning rate $\eta > 0$, the update minimizes the current linear loss plus a KL proximity term:

$$\mathbf{w}_{t+1}^{(v)} = \arg\min_{\mathbf{w} \in \Delta^{d_v}} \left\{ \eta \langle \mathbf{w}, \boldsymbol{\ell}_t^{(v)} \rangle + D_{\mathrm{KL}}(\mathbf{w} \| \mathbf{w}_t^{(v)}) \right\}. \tag{3}$$

Algorithm 1 RCT-PPR online retrieval

Input: Query q, entity graph G_{ent} and chunk associations, maximum rounds $T_{\max}$, temperature τ, fusion weight β, restart probability α
Output: Top-K chunks $\mathcal{C}_{final}$

Initialize each nonempty outgoing distribution uniformly: $w_{1,(u,v)} = 1/|\mathcal{N}(u)|$.
Obtain normalized seeds $\mathbf{s}_1$ through triple retrieval and LLM filtering.
Initialize context memory $\mathcal{M} \leftarrow \emptyset$.
for $t = 1, \ldots, T_{\max}$ **do**
  Construct $\mathbf{W}^{(t)}$ from the current local transition distributions.
  Compute $\boldsymbol{\pi}_t$ using Equation 1 with seeds $\mathbf{s}_t$.
  Map entity scores to candidate chunks $\mathcal{C}_t$; set $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{C}_t$.
  $\texttt{is_resolved}, \texttt{target_dir}, \mathbf{s}_{t+1} \leftarrow \mathrm{LLM_Controller}(\mathcal{M}, q)$.
  if $\texttt{is_resolved}$ **or** $t = T_{\max}$ **then**
    break
  end if
  Encode and normalize $\texttt{target_dir}$ to obtain $\mathbf{r}_t$.
  Update each nonempty outgoing distribution using Equation 4 with $\eta = 1/\tau$.
end for
Rank chunks in $\mathcal{M}$ by IDF-based entity aggregation and dense-score fusion.
return the top-K chunks $\mathcal{C}_{final}$.

With positive initialization, the stationarity condition and simplex constraint give

$$w_{t+1,j}^{(v)} = \frac{w_{t,j}^{(v)} \exp(\eta \langle \mathbf{e}_{v,j}, \mathbf{r}_t \rangle)}{\sum_{a=1}^{d_v} w_{t,a}^{(v)} \exp(\eta \langle \mathbf{e}_{v,a}, \mathbf{r}_t \rangle)}. \tag{4}$$

This is the standard MWU rule (Freund & Schapire, 1997; Hazan, 2016), with temperature $\tau = 1/\eta$. Local normalization preserves valid transition probabilities. Unlike assigning weights from only the latest target, MWU also retains the previous distribution: with uniform initialization and constant η, $w_{t+1,j}^{(v)} \propto \exp(\eta \sum_{s=1}^{t} \langle \mathbf{e}_{v,j}, \mathbf{r}_s \rangle)$. This accumulation can retain useful routing preferences, but it can also retain misleading feedback. In particular, a near-zero current alignment does not imply an unchanged normalized edge probability.

Local static regret. For uniform initialization, bounded losses as in Equation 2, and a fixed outgoing edge set, the standard exponential-weights analysis yields

$$\begin{aligned} R_T^{(v)} &:= \sum_{t=1}^{T} \langle \mathbf{w}_t^{(v)}, \boldsymbol{\ell}_t^{(v)} \rangle - \min_{\mathbf{u} \in \Delta^{d_v}} \sum_{t=1}^{T} \langle \mathbf{u}, \boldsymbol{\ell}_t^{(v)} \rangle \\ &\leq \frac{\log d_v}{\eta} + \eta T \leq \frac{\log d_{\max}}{\eta} + \eta T. \end{aligned} \tag{5}$$

Because the losses are linear, comparison with the best fixed distribution is equivalent to comparison with the best fixed outgoing edge. To obtain the bound, apply the log-sum-exp potential argument with losses in $[-1, 1]$: the one-round potential change is at most $-\eta \langle \mathbf{w}_t, \boldsymbol{\ell}_t \rangle + \eta^2/2$. Summing and comparing with a fixed edge gives the tighter bound $\log d_v/\eta + \eta T/2$, which implies Equation 5. The argument applies to the realized feedback sequence, including targets selected adaptively from the retrieval history.

Temperature and scope. For $d_{\max} > 1$, minimizing the displayed upper bound for a prescribed horizon T gives $\tau = \sqrt{T/\log d_{\max}}$ and $R_T^{(v)} = \mathcal{O}(\sqrt{T \log d_{\max}})$. This temperature minimizes a regret upper bound, not the QA error. The maximum retrieval budget supplies a horizon upper bound without requiring the true reasoning depth: with $\eta = \sqrt{\log d_{\max}/T_{\max}}$ and $H \leq T_{\max}$ feedback rounds, $R_H^{(v)} \leq \log d_{\max}/\eta + \eta H$. Nodes with a single outgoing edge have no transition choice and zero local regret; nodes with no outgoing edges require the PPR implementation's dangling-node rule.

Table 1: Statistics of the evaluation collections.

	HotpotQA	MuSiQue	2WikiMultihopQA
# of Documents (D)	300	200	300
# of Chunks (C)	30694	34845	28187
# of Queries (Q)	300	200	300
# of Entities (E)	178620	210580	181318
# of Triples (T)	187291	214761	175645

The guarantee controls cumulative semantic-alignment loss against a *fixed* comparator at one node. It does not control regret against a changing sequence of optimal edges, guarantee a globally optimal evidence path, or imply higher QA accuracy. In particular, it does not remove errors in the graph, embeddings, or controller. The degree dependence describes the local regret bound; query-time computation still depends on the graph size. Appendix D provides additional theoretical details.

4 Experiments

We evaluate answer quality, graph configurations, component ablations, and the effect of the retrieval-round budget.

4.1 Experimental Settings

Datasets. We evaluate on HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), and 2WikiMultihopQA (Ho et al., 2020), using the long-context QA setting derived from LongBench (Bai et al., 2024). Table 1 reports the document, chunk, query, entity, and triple counts in our evaluation collections.

Baselines and generation setting. We compare BM25, Contriever, and GTR with RAPTOR, LightRAG, HippoRAG, HippoRAG 2, and IRCoT. All main comparisons use GPT-4o for final answer generation from the top-5 selected chunks. This controls the final generator and number of chunks, but does not by itself match retrieval-stage computation, token counts, or latency. RCT-PPR uses a maximum of five retrieval rounds in the main comparison.

Metrics. We report exact match (EM), token-level F1, embedding-based answer similarity (Sem), and consensus-based factual correctness (CFC). F1 is the primary metric, and the average column is the unweighted mean across the three datasets. Sem complements lexical overlap but is not itself a factuality guarantee. CFC evaluates answer correctness under the assessment protocol in Appendix C. We treat CFC as an answer-level measure: a correct answer does not establish that the retriever recovered the complete supporting evidence chain. Direct evidence-retrieval quality and answer quality are distinct evaluation targets.

4.2 Results

4.2.1 Main Comparison

Table 2 shows that RCT-PPR achieves the highest F1 among the evaluated methods on each dataset, with an average of 53.3. HippoRAG 2 has the highest average F1 among the baselines at 46.9, giving a 6.4-point difference. Relative to HippoRAG 2, the dataset-level gains are 5.6 points on HotpotQA, 8.8 on MuSiQue, and 4.8 on 2Wiki. Baseline rankings differ by dataset: RAPTOR has the highest baseline F1 on HotpotQA, IRCoT on MuSiQue, and HippoRAG 2 on 2Wiki. The corresponding gains over these dataset-specific baselines are 4.1, 6.4, and 4.8 points.

These comparisons measure the full system, which includes iterative reasoning, seed updates, transition adaptation, and score fusion. They do not attribute the entire gain over HippoRAG 2 to the OCO update. We examine the components separately below.

Table 2: Answer quality with GPT-4o and the top-5 selected chunks. Average denotes the macro-average across datasets. Bold and underline mark the highest and second-highest reported values in each column. Retrieval-stage computation is not matched by this setting.

Method	HotpotQA				MuSiQue				2Wiki				Average			
	EM	F1	Sem	CFC	EM	F1	Sem	CFC	EM	F1	Sem	CFC	EM	F1	Sem	CFC
BM25	43.7	54.3	66.7	<u>59.8</u>	14.5	23.3	40.1	23.3	36.7	41.6	54.2	43.3	31.6	39.7	53.7	42.1
Contriever	37.8	48.9	64.2	51.7	11.0	16.9	39.1	17.5	35.2	41.9	54.0	43.3	28.0	35.9	52.4	37.5
GTR	43.1	55.1	68.9	57.8	18.5	26.8	46.2	27.8	42.0	50.4	62.4	51.5	34.5	44.1	59.2	45.7
RAPTOR	<u>45.3</u>	<u>55.9</u>	<u>70.1</u>	58.7	20.0	29.4	48.7	31.0	44.0	52.2	64.8	53.3	36.4	45.8	<u>61.2</u>	47.7
LightRAG	36.3	47.7	61.6	48.3	18.5	23.2	43.0	25.8	40.0	45.4	59.3	47.2	31.6	38.8	54.6	40.4
HippoRAG	34.1	47.1	59.4	46.7	15.0	20.0	40.5	19.5	39.7	45.1	58.2	46.3	29.6	37.4	52.7	37.5
HippoRAG 2	42.7	54.4	67.2	55.5	<u>26.3</u>	32.7	50.6	33.3	<u>46.6</u>	<u>53.7</u>	<u>65.1</u>	<u>56.2</u>	<u>38.5</u>	<u>46.9</u>	61.0	<u>48.3</u>
IRCoT	30.0	48.0	62.0	49.3	22.0	<u>35.1</u>	<u>53.1</u>	<u>37.5</u>	34.1	53.5	64.3	55.6	28.7	45.5	59.8	47.5
RCT-PPR(Ours)	**47.8**	**60.0**	**71.8**	**62.5**	**29.2**	**41.5**	**56.2**	**43.0**	**48.8**	**58.5**	**67.4**	**61.8**	**41.9**	**53.3**	**65.1**	**55.8**

Table 3: F1 under different diffusion graphs. Chunk associations remain available for answer-context selection in all variants.

Method	HotpotQA	MuSiQue	2Wiki	Avg
Directed, entities + chunks	58.7	41.1	57.2	52.3
Directed, entity-only	**60.0**	**41.5**	**58.5**	**53.3**
Undirected, entities + chunks	56.7	39.9	56.3	51.0
Undirected, entity-only	58.9	40.7	56.1	51.9

4.2.2 Ablation Analysis

Graph topology. Table 3 varies edge directionality and the inclusion of chunk nodes in diffusion. Directed entity-only diffusion gives the highest average F1, 53.3, compared with 52.3 when chunk nodes are included. In the undirected setting, the corresponding scores are 51.9 and 51.0. Thus, the default uses the directed entity subgraph for PPR and preserves entity-to-chunk associations for ranking. These results motivate separating relation-based diffusion from chunk-level context selection in our implementation.

Iterative retrieval and transition adaptation. Restricting retrieval to one round reduces average F1 from 53.3 to 48.1 (Table 4), a 5.2-point decrease. Removing the OCO-based transition update while retaining the iterative retrieval pipeline gives 51.3, a 2.0-point decrease. The latter comparison supports the contribution of the transition-update component within the iterative system. The two ablation differences quantify separate interventions in the full system and should not be added together. Removing semantic fallback or dynamic target directions yields average F1 scores of 52.2 and 51.5, respectively.

Retrieval-round budget. Table 5 varies the maximum number of retrieval rounds. Average F1 rises from 48.1 at one round to 51.9 at two and 52.2 at three; allowing five rounds gives 53.3. Most of the average improvement occurs within the first three rounds, with a further 1.1-point gain from three to five. Individual datasets need not improve monotonically with the budget. These results characterize the empirical accuracy–budget trade-off. Temperature sensitivity and early-stopping quality are separate from this budget sweep.

4.2.3 Inference Budget and Computational Cost

The online pipeline includes initial triple retrieval and LLM seed filtering, followed by repeated PPR retrieval and controller evaluation. Each continuing round also encodes a target direction and updates relation-edge weights. For H executed rounds, Algorithm 1 performs H PPR retrievals and H controller evaluations, in addition to initialization and final answer generation. Later controller prompts may contain more evidence, so equal call counts do not imply equal token budgets or latency.

With precomputed d-dimensional edge embeddings, a dense evaluation of the directional scores costs $\mathcal{O}(|\mathcal{E}_{rel}|d)$ per update; local normalization is linear in the number of relation edges. A sparse

Table 4: Component ablations (F1). The full system is included for reference. Differences are measured relative to the full system, rather than interpreted as additive contributions.

Method	HotpotQA	MuSiQue	2Wiki	Avg
Full RCT-PPR	**60.0**	**41.5**	**58.5**	**53.3**
Single retrieval round	55.5	35.6	53.1	48.1
w/o OCO-based update	58.4	40.5	55.0	51.3
w/o semantic fallback	58.9	41.3	56.3	52.2
w/o dynamic target direction	59.2	39.4	56.0	51.5

Table 5: F1 versus the maximum retrieval-round budget. The controller may stop before exhausting the budget.

$T_{\max}$	HotpotQA	MuSiQue	2Wiki	Avg
1	55.5	35.6	53.1	48.1
2	59.3	39.5	56.8	51.9
3	59.6	40.9	56.0	52.2
4	59.6	40.6	57.4	52.5
5	**60.0**	**41.5**	**58.5**	**53.3**

Table 6: Macro-average F1 in two LLM settings. The last column is GPT-4o-mini minus GPT-4o.

Method	GPT-4o	GPT-4o-mini	Difference
BM25	39.7	37.2	-2.5
Contriever	35.9	33.6	-2.3
GTR	44.1	41.4	-2.7
RAPTOR	45.8	43.0	-2.8
LightRAG	38.8	36.1	-2.7
HippoRAG	37.4	35.2	-2.2
HippoRAG 2	46.9	44.0	-2.9
IRCoT	45.5	41.8	-3.7
RCT-PPR	**53.3**	**50.2**	-3.1

PPR solver costs $\mathcal{O}(I(|\mathcal{E}_{rel}| + |\mathcal{V}_{ent}|))$ for I solver iterations. These are algorithmic costs, not wall-clock measurements. Offline extraction and indexing are separate preprocessing costs. The iteration ablation identifies lower-round configurations with competitive answer quality, but does not establish superiority under matched monetary cost or measured latency.

4.2.4 Sensitivity to the LLM Setting

We additionally evaluate a GPT-4o-mini setting under the same retrieval configuration. Table 6 reports macro-average F1 across the three datasets. RCT-PPR scores 50.2, compared with 44.0 for HippoRAG 2 and 41.8 for IRCoT. Relative to the GPT-4o setting, its score decreases by 3.1 points, while the advantage over HippoRAG 2 remains 6.2 points. These results support the persistence of the overall system advantage in this additional model setting. They do not separately identify the effects of controller quality and answer-generator quality.

4.2.5 Paired Comparison of Answer F1

We compare RCT-PPR with HippoRAG 2 and IRCoT using paired per-query F1 differences. Table 7 reports bootstrap confidence intervals for the mean difference and paired permutation-test p-values. All reported intervals exclude zero. These comparisons support the full-system improvements over the two baselines on the evaluated query sets; they do not establish the statistical significance of the OCO-component ablation or quantify variation across repeated model runs.

Table 7: Paired F1 differences between RCT-PPR and each baseline, in F1 points. Confidence intervals and p-values are reported separately for each dataset.

Baseline	Dataset	Difference	95% bootstrap CI	p
HippoRAG 2	HotpotQA	$+5.6$	$[+2.8, +8.5]$	0.001
	MuSiQue	$+8.8$	$[+4.5, +13.2]$	< 0.001
	2Wiki	$+4.8$	$[+1.9, +7.7]$	0.003
IRCoT	HotpotQA	$+12.0$	$[+8.4, +15.5]$	< 0.001
	MuSiQue	$+6.4$	$[+2.3, +10.6]$	0.002
	2Wiki	$+5.0$	$[+1.7, +8.3]$	0.004

Table 8: Primary failure categories in 50 manually inspected unsuccessful cases.

Assigned primary cause	Count	Share
Missing entity or relation in the KG	16	32%
Edge/entity embedding-related error	10	20%
Incorrect controller target or seed	12	24%
Insufficient iteration budget	4	8%
Final answer generation error	8	16%

4.2.6 Qualitative Failure Analysis

We manually inspect 50 failed RCT-PPR cases and assign one primary cause to each case (Table 8). Missing graph evidence is the most frequent assigned cause, followed by controller errors and embedding-related retrieval errors. Only four cases are assigned primarily to the iteration budget. These descriptive findings suggest that increasing the budget alone would not address many of the inspected failures. Because categories can interact and the analysis uses a limited sample, the counts are diagnostic observations rather than a causal decomposition or population-level failure estimate.

5 Conclusion

We presented RCT-PPR, an iterative graph retrieval framework that adapts PPR restart seeds and local transition distributions using LLM reasoning feedback. The transition update follows entropic mirror descent and admits a local static-regret interpretation for bounded semantic-alignment losses. On three LongBench multi-hop QA datasets, the full system achieves 53.3 average F1. The ablations show that multi-round retrieval improves on the single-round configuration and that the OCO-based transition update provides a further gain within the iterative pipeline.

Limitations. Performance depends on graph coverage, embedding quality, and the controller's ability to identify useful targets and stop appropriately. Accumulated transition weights can preserve incorrect feedback, and a local regret guarantee does not prevent such errors or ensure evidence-chain recovery. Iterative reasoning adds query-time computation beyond single-pass retrieval. The present main experiments use one LLM setting and do not establish robustness to controlled graph corruption or efficiency under matched token and latency budgets. Future evaluation should distinguish controller and generator effects, measure supporting-evidence coverage, and compare transition rules under the same graph, controller, and inference budget.

AI Use Statement

Generative AI was used in manuscript revision to assist with language editing, organization, literature discussion, consistency checks, refinement of mathematical claims and experimental interpretations, and assistance with a standard regret-bound derivation. The authors are responsible for the final manuscript and its claims. LLMs are also components of the proposed retrieval and generation pipeline, as described in the methodology.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2023.

Baruch Awerbuch and Robert D Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pp. 45–53, 2004.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 3119–3137, 2024.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

Zheng Chu, Jingchang Chen, Qianglong Chen, Haotian Wang, Kun Zhu, Xiyuan Du, Weijiang Yu, Ming Liu, and Bing Qin. Beamaggr: Beam aggregation reasoning over multi-source knowledge for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1229–1248, 2024.

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

Jinyuan Fang, Zaiqiao Meng, and Craig Macdonald. Kirag: Knowledge-driven iterative retriever for enhancing retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18969–18985, 2025.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tian Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2(3), 2024.

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*, 2025.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, et al. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9844–9855, 2022.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.

Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc, 2009.

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

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 10014–10037, 2023.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

APPENDICES

The supplementary material contains experimental configurations in Appendix A, prompt descriptions in Appendix B, evaluation details in Appendix C, and theoretical derivations in Appendix D.

A ADDITIONAL EXPERIMENTAL DETAILS

A.1 DATASETS AND PREPROCESSING

We evaluate on the HotpotQA, MuSiQue, and 2WikiMultihopQA tasks derived from LongBench (Bai et al., 2024). These tasks require reasoning over evidence in multiple Wikipedia documents. HotpotQA includes questions linking supporting facts across documents (Yang et al., 2018); MuSiQue composes single-hop questions into multi-hop questions (Trivedi et al., 2022); and 2WikiMultihopQA uses structured knowledge and question templates to make reasoning steps explicit (Ho et al., 2020). Table 1 in the main text reports the sizes of our evaluation collections.

Long-context inputs. The evaluated contexts contain supporting passages together with distractor passages. The construction described for these collections expands the supporting context with longer Wikipedia passages, adds distractors, and shuffles passage order before chunking. Passage shuffling varies the positions of evidence but does not guarantee the absence of positional effects. The evaluation measures retrieval within the assembled contexts; it should be distinguished from retrieval over an unrestricted Wikipedia collection.

Overlapping chunks. Repeated whitespace is normalized, and each assembled context is divided into overlapping windows. Let L denote the maximum window length and O the overlap, measured in the same unit. Consecutive window starts are separated by $L - O$; overlap and stride are therefore different parameters. Chunk identifiers combine the parent context identifier with a sequential window index. The overlap is intended to retain local context near boundaries, although a reasoning chain may still span multiple chunks.

A.2 GRAPH CONSTRUCTION AND ONLINE RETRIEVAL

Information extraction. For each chunk, `gpt-4o-mini` extracts entities, semantic types, and directed relations in a JSON format. The extraction template is shown in Figure 4. Chunks are processed independently using `ThreadPoolExecutor`, and their extracted records are merged into the stored graph.

Stored graph and diffusion graph. The stored graph contains two node types: entity nodes $\mathcal{V}_{ent}$ and chunk nodes $\mathcal{V}_{chk}$. Directed entity–entity edges represent extracted relations. Chunk–entity associations record which entities occur in which chunks, using the stored `contains` and/or `mentioned_in` relations. These associations support the mapping from entity relevance to source text. In the default configuration of RCT-PPR, diffusion uses the directed entity-relation subgraph; chunk associations are used during context selection. This distinction matches the entity-only configuration in Table 3.

Relation embeddings. Each relation edge is represented as *"[Head] [Relation] [Tail]"* and encoded with `all-MiniLM-L6-v2`. Cosine similarity between the relation representation and a target-direction representation supplies the alignment signal. The theoretical analysis uses unit-normalized, nonzero representations, so the inner product equals cosine similarity. The text encoder used for a target direction must be compatible with the edge encoder.

Seeds, retrieval rounds, and final context. Query–triple similarity selects five candidate triples. LLM seed filtering identifies entities for the initial restart distribution, after which the controller revises the retrieval focus using accumulated evidence. The main configuration allows up to five retrieval rounds and supplies five selected chunks to the final answer generator. The exact ordering of PPR retrieval and transition updates follows Algorithm 1. In that algorithm, the first PPR call uses the initialized weights, and feedback from a retrieval round changes the weights for the next round.

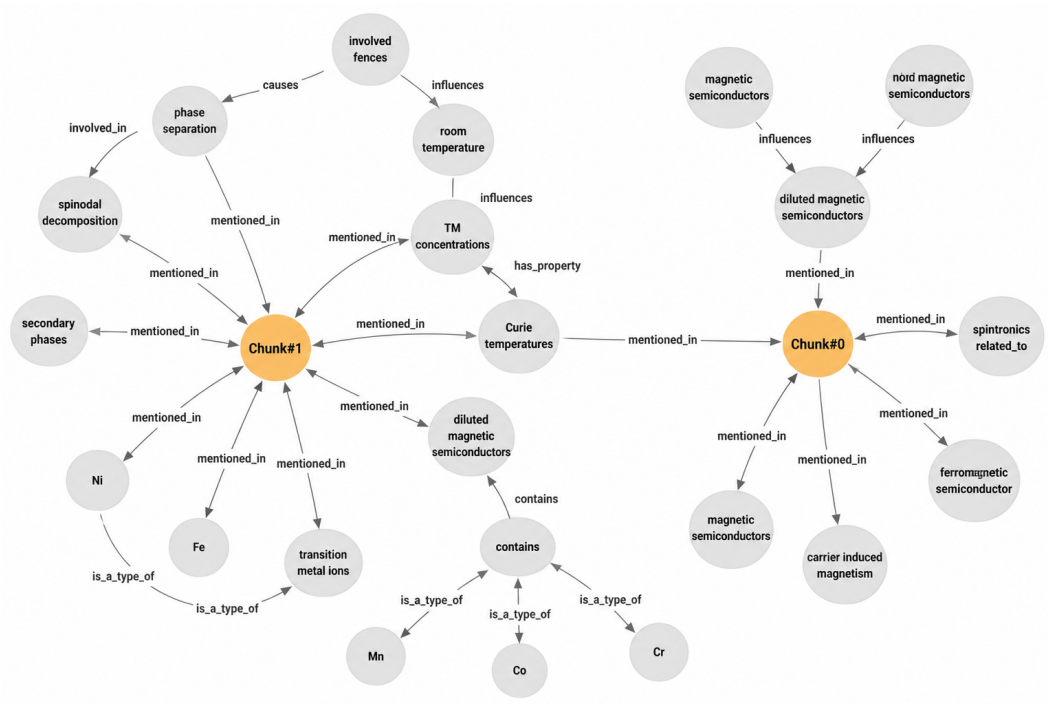

Figure 3: Illustration of the stored entity–chunk graph. Entity relations support graph diffusion, while chunk associations connect structured records to source text. The default diffusion configuration excludes chunk nodes.

Answer generation. The selected context is supplied to `gpt-4o` using decoding temperature zero. This setting reduces sampling variability, but does not guarantee identical responses across API versions or executions. A common answer model controls one component of the comparison; answer quality still depends on both the retrieved evidence and the generator.

PPR convention. Throughout the mathematical analysis, α denotes the *restart* probability, so $\boldsymbol{\pi} = (1-\alpha)\mathbf{W}\boldsymbol{\pi} + \alpha\mathbf{s}$. Some implementations instead expose the continuation or damping factor, denoted here by $\rho = 1-\alpha$. A damping factor $\rho = 0.85$ therefore corresponds to restart probability $\alpha = 0.15$. This convention prevents conflating the reported baseline damping setting with the restart parameter in the equations.

A.3 Baseline Configurations

The following descriptions specify the configurations evaluated in this study. Their retrieval backends, encoders, and graph choices are part of the comparison and should be distinguished from claims about every implementation of the corresponding method.

BM25. We use BM25Okapi with whitespace tokenization for chunks and queries.

Contriever and GTR. Contriever uses the `facebook/contriever` checkpoint through SentenceTransformers. GTR uses `sentence-transformers/gtr-t5-base`. Both encode queries and chunks for dense retrieval.

RAPTOR. The hierarchy is built from original chunks using Gaussian mixture clustering after PCA to ten dimensions. The cluster-count rule is based on $N/4$ for a layer containing N nodes. `gpt-4o-mini` summarizes clusters, and construction stops when a layer contains at most ten nodes. Retrieval compares the query with all indexed levels, including original chunks and summaries, rather than traversing one branch of the hierarchy.

Triples Extraction Prompt

SYSTEM

You are the information extraction engine, only outputting strict JSON.

USER

Extract entities and relations from the following text and return JSON:
- entities: [{{ "name": entity, "type": type }}]
- relations: [{{ "head": head entity, "tail": tail entity, "type": relation type }}]
Text: {chunk}

Figure 4: JSON-format extraction template for entities and directed relations.

LightRAG. Our configuration loads precomputed graphs, chunks, and entity records, constructs chunk and entity vector indices, and uses hybrid retrieval through these stores. A key-value store maps retrieved identifiers back to text. Prebuilt indices are reused across queries when available, and answer generation uses temperature zero.

IRCoT. The evaluated configuration uses GPT-4o for reasoning and BM25 for retrieval, with at most five rounds. Each round retrieves five chunks and appends them to context memory. The model produces either a new query prefixed with `Search:` or a `Final Answer`. Its F1 scores are 48.0 on HotpotQA, 35.1 on MuSiQue, and 53.5 on 2WikiMultihopQA. The iterative context can differ in size from a single-pass context, so iteration count alone does not determine inference cost.

HippoRAG. In the evaluated configuration, entity representations use `all-MiniLM-L6-v2`. The five entities with highest query cosine similarity form the seed set. PPR uses a continuation factor $\rho = 0.85$. Chunk relevance is obtained by aggregating associated entity scores, and five chunks are selected. The solver's initial iterate is distinct from its query-dependent restart distribution.

HippoRAG 2. Triple embeddings use `all-MiniLM-L6-v2`. Five recalled triples are filtered by `gpt-4o` at temperature zero to identify seed entities, whose weights are derived from their originating triple similarities. The evaluated configuration uses PPR with continuation factor $\rho = 0.85$ and propagation restricted to entity–chunk connections, followed by selection of five chunks.

B Prompts and Templates

LLMs serve different roles in the pipeline. Offline entity/relation extraction uses `gpt-4o-mini`; the online reasoning controller and final answer generator use `gpt-4o` in the main setting. Thus, the statement that every prompt uses GPT-4o would conflate offline construction with online inference. The prompt figures below retain the templates supplied with the implementation.

B.1 Entity and Relation Extraction

The extraction prompt requests structured records from each source chunk. A JSON output constraint facilitates parsing but does not guarantee valid syntax or factual extraction. The extracted records determine which relations can subsequently be traversed.

B.2 Reasoning and Routing

The controller compares retrieved context with the question and reasoning history, predicts whether the evidence is sufficient, and proposes entities and a `next_target_direction`. It is instructed

Evaluator Prompt

Question: "{query}"

Previous Reasoning History:
{reasoning_history if reasoning_history else 'None'}

Newly Retrieved Context:
{context}

Task:
1. Read the Previous Reasoning History and the Newly Retrieved Context.
2. Determine if you now have enough information to fully answer the Question correctly.
3. If YES: Set 'is_resolved' to true, summarize the final reasoning in 'updated_reasoning'.
   'next_seeds' can be empty.
4. If NO: Synthesize what you have learned so far into 'updated_reasoning' (e.g.,
   'We know A is related to B, so we need to find C').
   Then, determine the set of core entities that form the current reasoning chain and
   will drive the next search step.
   Your 'next_seeds' MUST include all critical entities driving the logic
   (e.g., if the logic flows from A -> B -> C, output ["A", "B", "C"]).
   IMPORTANT: Extract these entities precisely as they appear in the Context or
   Reasoning History to ensure they match nodes in the underlying Knowledge Graph.
5. Output strictly in JSON format:

```
{
  "is_resolved": true/false,
  "updated_reasoning": "<your updated step-by-step reasoning>",
  "next_target_direction": "<A short sentence describing exactly what relation or entity
    you are looking for next, e.g., 'Looking for the CEO of the company' or
    'Searching for A's relationship with B'>",
  "next_seeds": ["<entity_A>", "<entity_B>"]
}
```

Figure 5: Controller prompt for assessing retrieved evidence and proposing the next entities and target direction.

to identify entity mentions precisely as they occur in the text. This instruction supports matching to graph records, but entity linking still requires an implementation-level mapping from a textual mention to a node identifier.

The target direction is a text string, which is encoded as $\mathbf{r}_t$. It is *not itself a scalar loss*: edge-level losses are obtained as $\ell_{t,j}^{(v)} = -\langle \mathbf{e}_{v,j}, \mathbf{r}_t \rangle$ after normalization. The stopping flag is likewise a model judgment rather than a proof that the evidence chain is complete.

B.3 Answer Generation

The answer prompt requests a concise response grounded in the selected chunks. Exact match is an evaluation metric; an extractive instruction does not guarantee that the model will produce an exact reference string. GPT-4o decoding temperature is set to zero for the online reasoning and answer-generation prompts.

C Evaluation Details

C.1 Automated Answer Metrics

We report exact match, token-level F1, and embedding-based semantic similarity (Sem), alongside the human-assessed CFC metric below. F1 is the primary answer-quality measure. These metrics assess generated answers and should be distinguished from recall of supporting documents or evidence chains. Macro-averages give equal weight to each dataset.

Answer Generation Prompt

Read the following text and answer briefly.
{context_text}

Now, answer the following question based on the above text, only give me the answer and do not output any other words.

Question: {query}

Answer:

Figure 6: Prompt used to generate the final answer from selected evidence.

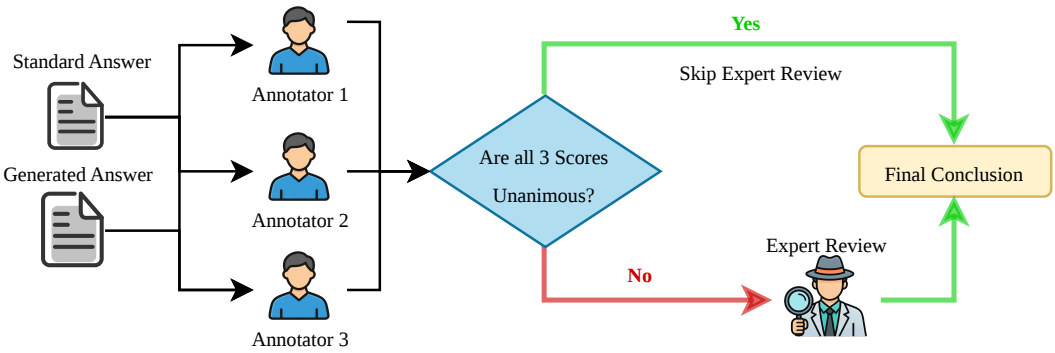

Figure 7: Human-review workflow for CFC: independent scores are compared, and disagreements are referred to an expert adjudicator.

C.2 Consensus-Based Factual Correctness

We use the name **Consensus-based Factual Correctness (CFC)** consistently for the supplementary answer-correctness assessment. Its item-level scores are 0 (incorrect), 0.5 (partially correct), and 1 (correct). A correct answer is not sufficient evidence that all supporting facts were retrieved: the generator can also rely on prior knowledge or incomplete evidence.

Human scoring and adjudication. For items entering human review, three annotators assign independent scores. If all three scores agree, that score is retained. Otherwise, an expert reviews the disputed item and determines its final score; majority voting alone does not resolve a disagreement. Answer correctness is assessed in relation to the question and supporting context. The human-review workflow is illustrated in Figure 7.

Aggregation. If $c_i \in \{0, 0.5, 1\}$ is the finalized score for an evaluated answer and n answers are assessed, the percentage-scale score is

$$\mathrm{CFC} = \frac{100}{n} \sum_{i=1}^{n} c_i. \tag{6}$$

The denominator and any sampling of answers must be distinguished from the total number of dataset queries. An adjudicated score summarizes the implemented annotation protocol, not an error-free measure of factual truth.

Answers containing extra entities. Consider the supplied evaluation example: the question asks for the spouse of the director of *Rodney Steps In*, the reference is *Ivy Duke*, and the prediction is *Ivy Duke and Dorothy Batley*. The prediction contains the reference string, but string inclusion alone cannot determine whether the complete answer is correct. Review must check the additional assertion against the question and supporting evidence. Conversely, absence from the reference string alone is not proof that an additional assertion is false. Partial credit requires an explicit annotation rule and cannot be inferred merely from the presence of the reference entity.

We therefore distinguish exact string matching, lexical overlap, and adjudicated factual correctness. Fixed EM/F1 implementations apply their respective deterministic rules; ambiguity in an answer's factual interpretation does not imply that the same metric arbitrarily assigns 0, 0.5, or 1. The case illustrates the need to document CFC guidelines rather than establishing a new scoring decision for that example.

D Theoretical Details and Proofs

We separate three mathematical statements: how individual walks contribute to PPR, what can be concluded *conditional on* updating the restart distribution, and the local regret of multiplicative-weights transition adaptation. These statements have different assumptions and do not together constitute an end-to-end QA guarantee.

D.1 PPR Walk Contributions and Restart Distributions

Let $\mathbf{W}$ be a nonnegative column-stochastic matrix and $\mathbf{s}$ a probability vector. With restart probability $\alpha \in (0, 1)$, PPR satisfies

$$\boldsymbol{\pi} = (1-\alpha)\mathbf{W}\boldsymbol{\pi} + \alpha\mathbf{s}. \tag{7}$$

Because the spectral radius of $(1-\alpha)\mathbf{W}$ is less than one,

$$\boldsymbol{\pi} = \alpha\big(\mathbf{I} - (1-\alpha)\mathbf{W}\big)^{-1}\mathbf{s} = \alpha\sum_{k=0}^{\infty}(1-\alpha)^k\mathbf{W}^k\mathbf{s}. \tag{8}$$

This is the unique normalized nonnegative solution. The stochastic-matrix assumption includes whatever completion rule the implementation uses for dangling nodes.

Proposition 1: Contribution of a specified walk. For a one-hot restart at v_0, the contribution of a particular length-H walk $\mathcal{P} = (v_0, \ldots, v_H)$ to the $k = H$ term of Equation equation 8 is

$$\pi_{\mathcal{P}}(v_H) = \alpha(1-\alpha)^H \prod_{i=0}^{H-1} W_{v_{i+1},v_i}. \tag{9}$$

Proof. A walk following these transitions has probability $\prod_i W_{v_{i+1},v_i}$. The length-H term in the PPR series multiplies this probability by $\alpha(1-\alpha)^H$. For uniform outgoing transitions, $W_{v_{i+1},v_i} = 1/d_i$, so

$$\pi_{\mathcal{P}}(v_H) = \alpha(1-\alpha)^H \prod_{i=0}^{H-1} \frac{1}{d_i}. \tag{10}$$

This establishes the geometric length discount and the multiplicative branching factors for that walk. The total stationary mass $\pi(v_H)$ sums contributions from all admissible walks and lengths; Equation equation 10 is not an upper bound on that total.

A shortest-distance upper bound. If the shortest positive-probability path from the support of $\mathbf{s}$ to a node x has length h, then $(\mathbf{W}^k\mathbf{s})_x = 0$ for $k < h$ and $(\mathbf{W}^k\mathbf{s})_x \le 1$ otherwise. Consequently,

$$\pi(x) \le \alpha\sum_{k=h}^{\infty}(1-\alpha)^k = (1-\alpha)^h. \tag{11}$$

The distance is measured in the transition graph after any dangling-node completion. This bound captures a possible distance-related limitation without identifying one walk's contribution with the node's entire score.

Proposition 2: Conditional benefit of re-anchoring. At a retrieval round t, suppose the restart distribution assigns mass $\mathbf{s}_t(u) \geq \gamma > 0$ to an intermediate node u. Let $(u = u_0, u_1, \ldots, u_L = x)$ be a length-L walk in the current transition matrix. Then

$$\pi_t(x) \geq \alpha\gamma(1-\alpha)^L \prod_{i=0}^{L-1} W^{(t)}_{u_{i+1},u_i}. \tag{12}$$

Proof. Retain only the specified nonnegative walk contribution from the $k = L$ term of the PPR series for $\mathbf{W}^{(t)}$ and $\mathbf{s}_t$. Its restart mass is at least γ, which gives the bound. In the special case where the target itself is assigned restart mass,

$$\pi_t(x) \geq \alpha \mathbf{s}_t(x). \tag{13}$$

If $\mathbf{s}_t(x) = 1$, then $\pi_t(x) \geq \alpha$.

This result describes the effect of a *given* restart distribution. It does not prove that a controller can discover the correct intermediate node or identify the target in advance. A lower bound independent of the original reasoning depth additionally requires a lower bound on γ, a bounded remaining length L, and suitable lower bounds on the relevant transition probabilities. Moreover, positive target mass alone does not guarantee a top-K rank or successful chunk selection. Re-anchoring benefits any PPR method with the same restart distribution; it is separate from the contribution of MWU. Thus, the two propositions do not establish an unconditional exponential separation between retrieval systems.

D.2 Derivation of the Local MWU Update

Consider a node v with a fixed outgoing edge set of size $d_v \geq 1$. Let $\mathbf{w}^{(v)}_t \in \Delta^{d_v}$ be its outgoing distribution before feedback at round t. For normalized embeddings, define

$$\ell^{(v)}_{t,j} = -\langle \mathbf{e}_{v,j}, \mathbf{r}_t \rangle \in [-1, 1], \qquad L^{(v)}_t(\mathbf{w}) = \langle \mathbf{w}, \boldsymbol{\ell}^{(v)}_t \rangle. \tag{14}$$

In the feedback ordering of Algorithm 1, the controller obtains its target after retrieval using the current weights, and the target is used to update the next round's weights. Losses may depend on the previous retrieval trajectory. Future targets need not be known.

For a learning rate $\eta > 0$, the entropic mirror-descent step is

$$\mathbf{w}^{(v)}_{t+1} = \arg\min_{\mathbf{w} \in \Delta^{d_v}} \left\{ \eta \langle \mathbf{w}, \boldsymbol{\ell}^{(v)}_t \rangle + \sum_{j=1}^{d_v} w_j \log \frac{w_j}{w^{(v)}_{t,j}} \right\}. \tag{15}$$

Assume $w^{(v)}_{t,j} > 0$ for all outgoing edges. The KL divergence uses the convention $0 \log 0 = 0$; it is finite at a zero component of the first argument. Positive initialization and finite losses yield the strictly positive optimum derived below, rather than excluding the boundary merely by invoking the logarithm.

Introducing a multiplier λ for $\sum_j w_j = 1$ gives

$$\mathcal{L}(\mathbf{w}, \lambda) = \eta \sum_j w_j \ell^{(v)}_{t,j} + \sum_j w_j \log \frac{w_j}{w^{(v)}_{t,j}} + \lambda \left(\sum_j w_j - 1 \right). \tag{16}$$

Stationarity at an interior optimum requires

$$\eta \ell^{(v)}_{t,j} + \log \frac{w_j}{w^{(v)}_{t,j}} + 1 + \lambda = 0. \tag{17}$$

Consequently, $w_j = w^{(v)}_{t,j} \exp(-\eta \ell^{(v)}_{t,j}) \exp(-1 - \lambda)$. Define

$$Z^{(v)}_t = \sum_{a=1}^{d_v} w^{(v)}_{t,a} \exp(-\eta \ell^{(v)}_{t,a}). \tag{18}$$

The equality constraint gives $\exp(-1-\lambda) = 1/Z_t^{(v)}$, and hence

$$w_{t+1,j}^{(v)} = \frac{w_{t,j}^{(v)} \exp(-\eta \ell_{t,j}^{(v)})}{Z_t^{(v)}} = \frac{w_{t,j}^{(v)} \exp(\langle \mathbf{e}_{v,j}, \mathbf{r}_t \rangle / \tau)}{\sum_a w_{t,a}^{(v)} \exp(\langle \mathbf{e}_{v,a}, \mathbf{r}_t \rangle / \tau)}, \qquad \tau = \eta^{-1}. \tag{19}$$

The objective in Equation equation 15 differs from $D_{\mathrm{KL}}(\mathbf{w} \| \mathbf{w}_{t+1}^{(v)})$ by a constant, so this normalized positive vector is its unique minimizer. The update preserves the existing edge support and does not add missing graph relations.

Accumulated feedback and changing targets. For constant η, the recursion yields

$$w_{t+1,j}^{(v)} \propto w_{1,j}^{(v)} \exp\left(\eta \sum_{s=1}^{t} \langle \mathbf{e}_{v,j}, \mathbf{r}_s \rangle \right). \tag{20}$$

This distinguishes cumulative MWU from recomputing a softmax using only the current target. It can preserve past preferences, including incorrect ones. Even if an edge has zero alignment with the latest target, its normalized probability can change because the denominator changes. Orthogonality between selected target directions alone does not prove absence of interference between reasoning rounds.

D.3 Local Static-Regret Bound

We now use a constant $\eta > 0$ and any positive initial distribution. Fix a node and omit its superscript. For a fixed comparator $\mathbf{u} \in \Delta^{d_v}$, the update implies

$$\begin{aligned} D_{\mathrm{KL}}(\mathbf{u} \| \mathbf{w}_t) - D_{\mathrm{KL}}(\mathbf{u} \| \mathbf{w}_{t+1}) &= \sum_j u_j \log \frac{w_{t+1,j}}{w_{t,j}} \\ &= -\eta \langle \mathbf{u}, \boldsymbol{\ell}_t \rangle - \log Z_t. \end{aligned} \tag{21}$$

Log-partition bound. Define $f(a) = \log \sum_j w_{t,j} \exp(-a \ell_{t,j})$. Then $f(0) = 0$, $f'(0) = -\langle \mathbf{w}_t, \boldsymbol{\ell}_t \rangle$, and $f''(a)$ is the variance of $\ell_{t,j}$ under the exponentially tilted distribution. Since $\ell_{t,j} \in [-1, 1]$, this variance is at most one. Integrating the second derivative over $[0, \eta]$ gives

$$\log Z_t = f(\eta) \le -\eta \langle \mathbf{w}_t, \boldsymbol{\ell}_t \rangle + \frac{\eta^2}{2}. \tag{22}$$

This bound holds for every $\eta > 0$. In particular, it does not require the additional condition $\eta \le 1$ that would arise from using $e^x \le 1 + x + x^2$ only for $|x| \le 1$.

Combining Equations equation 21 and equation 22 yields

$$\eta \langle \mathbf{w}_t - \mathbf{u}, \boldsymbol{\ell}_t \rangle \le D_{\mathrm{KL}}(\mathbf{u} \| \mathbf{w}_t) - D_{\mathrm{KL}}(\mathbf{u} \| \mathbf{w}_{t+1}) + \frac{\eta^2}{2}. \tag{23}$$

Summing over T feedback rounds telescopes the KL terms. Since the final KL divergence is non-negative,

$$\sum_{t=1}^{T} \langle \mathbf{w}_t - \mathbf{u}, \boldsymbol{\ell}_t \rangle \le \frac{D_{\mathrm{KL}}(\mathbf{u} \| \mathbf{w}_1)}{\eta} + \frac{\eta T}{2}. \tag{24}$$

For uniform initialization, $D_{\mathrm{KL}}(\mathbf{u} \| \mathbf{w}_1) = \log d_v + \sum_j u_j \log u_j \le \log d_v$. Thus,

$$\begin{aligned} R_T^{(v)} &:= \sum_{t=1}^{T} \langle \mathbf{w}_t^{(v)}, \boldsymbol{\ell}_t^{(v)} \rangle - \min_{\mathbf{u} \in \Delta^{d_v}} \sum_{t=1}^{T} \langle \mathbf{u}, \boldsymbol{\ell}_t^{(v)} \rangle \\ &\le \frac{\log d_v}{\eta} + \frac{\eta T}{2} \le \frac{\log d_v}{\eta} + \eta T. \end{aligned} \tag{25}$$

Because the loss is linear, the best fixed distribution can be chosen as a point mass on a best fixed outgoing edge. The bound holds along the realized sequence of losses, including when targets depend on retrieval history. The comparator is evaluated on that same realized sequence, not on the different controller outputs that might have occurred under another retrieval policy.

Initialization and scope. A nonuniform positive initialization uses Equation equation 24; the uniform $\log d_v$ bound should not be silently retained after changing the initialization. A query-directed update can alternatively be treated as an explicitly indexed initial feedback step, provided that the round indexing in the algorithm and analysis is changed consistently. Nodes with one outgoing edge have zero local regret. Nodes with no outgoing edges have no such local simplex and are handled by the PPR completion rule.

The result concerns a node's cumulative proxy loss against a fixed comparator. It does not bound dynamic regret against a sequence of changing optimal edges, certify controller correctness, or guarantee retrieval of an evidence chain. Its dependence on local degree is also not a statement that full-graph computation is independent of the number of nodes or edges.

D.4 Temperature Choice and the Retrieval Budget

To match the main-text expression, consider the looser upper bound $B(\eta) = \log d_{\max}/\eta + \eta T$, where $d_{\max} > 1$. Differentiation gives

$$\eta_B = \sqrt{\frac{\log d_{\max}}{T}}, \qquad \tau_B = \sqrt{\frac{T}{\log d_{\max}}}, \qquad B(\eta_B) = 2\sqrt{T \log d_{\max}}. \tag{26}$$

This is a shared temperature obtained from the maximum-degree bound. Optimizing a separate local bound would instead produce a node-specific temperature involving d_v; these are distinct design choices. Also, optimizing the tighter bound in Equation equation 25 changes the constant, giving $\eta = \sqrt{2 \log d_{\max}/T}$. Consequently, the theory does not uniquely determine a universally optimal QA temperature.

Here T counts feedback rounds and is not the number of semantic hops in a question. For a prescribed maximum retrieval budget $T_{\max}$, setting $\eta = \sqrt{\log d_{\max}/T_{\max}}$ gives, for any realized feedback horizon $H \le T_{\max}$,

$$R_H^{(v)} \le \frac{\log d_v}{\eta} + \frac{\eta H}{2} \le \frac{\log d_{\max}}{\eta} + \eta T_{\max}. \tag{27}$$

This supplies a horizon upper bound without knowing the true reasoning depth. For the provided algorithm, only feedback used before a subsequent PPR call affects retrieval; the final unused update can be omitted. A fixed temperature selected from $T_{\max}$ remains fixed within the query even though controller targets change across rounds. If $d_{\max} = 1$, there is no local transition choice and this temperature formula is unnecessary.

The iteration-budget ablation measures answer quality as the number of allowed rounds changes. It does not compare temperature schedules or establish superiority to grid-search tuning. Such a claim requires a separate experiment controlling the retrieval budget and all other components.