RL-Enhanced ProRAG: A Proactive Retrieval-Augmented Generation Framework for Multi-Hop Reasoning

Xuewei Luo[1], Yujia Huo[1,2*], Yongbin Qin[1], Fujian Feng[1], Gan Liu[1], Dawen Xia[3], Wong Derek F[4]
[1]School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang, China
[2]Guizhou Provincial Key Laboratory of Applied Mathematics and Computing Power & Algorithms, Guizhou University, Guiyang, China
[3]College of Microelectronics and Artificial Intelligence, College of Big Data Engineering, Kaili University, Kaili, China
[4]The University of Macau, Macau, China
{xueweiluo, huo.yujia, ybqin, fujian_feng, liugan}@gzmu.edu.cn, dawenxia@kluniv.edu.cn, derekfw@umac.mo
*Corresponding author.

***Abstract*—Large Language Models often struggle with factual inaccuracies and knowledge limitations in complex, knowledge-intensive tasks. While Retrieval-Augmented Generation (RAG) mitigates these issues, existing dynamic RAG methods suffer from "retrieval myopia" in multi-hop reasoning, leading to redundant retrievals, incomplete queries, and contextual fragmentation.**

To address these challenges, we propose RL-Enhanced ProRAG, a novel framework that integrates Reinforcement Learning (RL) with a proactive retrieval paradigm. The framework features two synergistic modules: a Proactive Information Need Prediction (PINP) module for global retrieval planning, and a Core Entity Graph-based Query Expansion (CEQE) module for semantic query enhancement. A lightweight RL controller dynamically calibrates the confidence thresholds of PINP and the filtering strategies of CEQE based on cross-encoder relevance signals, enabling adaptive and precise retrieval.

Extensive experiments on four multi-hop reasoning benchmarks (2WikiMultihopQA, HotpotQA, StrategyQA, IIRC) show that RL-Enhanced ProRAG outperforms state-of-the-art dynamic RAG methods by $3-5\%$ in accuracy and F1 score, with relative improvements reaching up to 26.2% on high-complexity queries. Ablation studies confirm the critical role of the RL component, with its removal causing a performance drop of $1.5-2.3\%$.

Our contributions are threefold: (1) We propose the first unified framework that integrates RL with proactive retrieval to systematically address retrieval myopia; (2) We design a lightweight RL mechanism leveraging cross-encoder rewards for precision optimization; and (3) We provide comprehensive empirical validation demonstrating superior effectiveness, robustness, and balanced efficiency across diverse reasoning scenarios.

***Index Terms*—Retrieval-Augmented Generation, Multi-hop Reasoning, Reinforcement Learning, Proactive Retrieval, Large Language Models.**

I. Introduction

The accelerated development of large language models (LLMs) has precipitated a revolutionary period in natural language processing (NLP), characterised by extraordinary advancements in text generation and comprehension, and reasoning [1], [3]. Representative models such as the GPT series [2], LLaMA [4], and GLM [5], [6] have achieved state-of-the-art performance across a wide range of tasks. Despite these achievements, LLMs continue to be fundamentally constrained by their parametric knowledge, which is inherently incomplete, potentially outdated, and susceptible to hallucinations, producing fluent but factually incorrect content [7]. These limitations become particularly evident in knowledge-intensive scenarios that require factual accuracy and multi-step reasoning.

Retrieval-Augmented Generation (RAG) has emerged as a promising paradigm to alleviate these issues by augmenting LLMs with external knowledge sources [8]. By conditioning generation on retrieved documents, RAG improves factual grounding, reduces hallucinations, and enables access to up-to-date information without retraining the model [9]. However, the conventional "Naive RAG" paradigm relies on a single, static retrieval step, making its performance highly sensitive to retrieval quality. In practice, irrelevant or noisy passages may even degrade generation quality compared to using the LLM alone [10], [11].

In order to address the limitations of single-shot retrieval, recent work has explored Dynamic RAG frameworks that interleave retrieval and generation. The objective of these methods is to ascertain the optimal timing and formulation of retrieval queries during the generation process. Representative approaches include FLARE [12], which triggers retrieval based on token-level uncertainty; Self-RAG [13], which enables self-reflection on retrieval needs; and Graph RAG [14], which leverages document-level knowledge graphs for structured reasoning. Despite their effectiveness, existing dynamic RAG methods remain fundamentally reactive. Retrieval decisions are primarily driven by local uncertainty signals or recent generation context, without explicit consideration of the global information requirements of the entire reasoning process.

This reactive nature gives rise to a critical limitation in

This work was supported by the National Natural Science Foundation of China (Grant No. 62266013), Guizhou Provincial Basic Research Program (Grant No. MS[2026]276), Guizhou Provincial Key Laboratory of Applied Mathematics and Computing Power & Algorithms (Qiankehe Platform ZSYS[2025]039), and Guizhou Provincial Science and Technology Major Project (Qiankehe Major[2025]034).

complex multi-hop reasoning tasks, which is referred to as retrieval myopia. Retrieval myopia is defined as the inability of dynamic RAG systems to plan retrieval actions based on global, multi-step information needs. Consequently, such systems frequently encounter three interconnected challenges: (1) **Redundant retrievals**, where fixed-interval or sentence-level triggers lead to unnecessary searches [15], [16]; (2) **Incomplete queries**, as queries derived from narrow, local context fail to capture the full information requirements of multi-hop reasoning chains [17]; and (3) **Context fragmentation**, where iteratively appended snippets form a disjointed context that hinders coherent information synthesis, often manifesting as the "lost in the middle" problem [18]. Together, these issues degrade both retrieval efficiency and downstream answer quality.

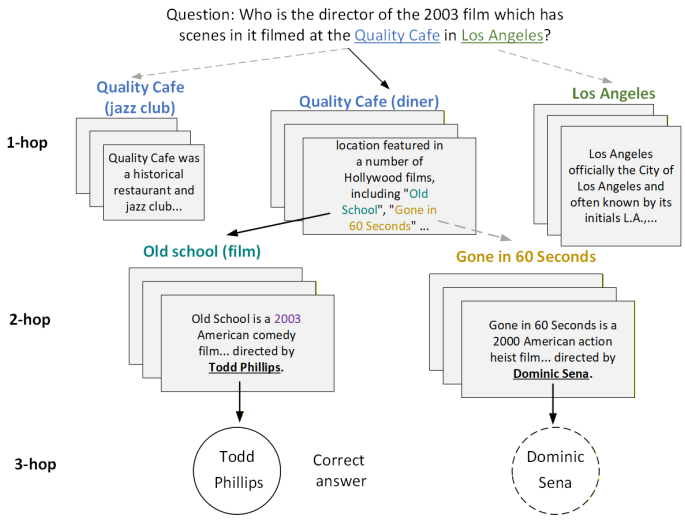


Fig. 1: Example of a multi-hop reasoning question

In order to address these challenges, a unified framework has been proposed, which combines proactive retrieval planning with RL-driven precision calibration. This framework is referred to as RL-Enhanced ProRAG (Reinforcement Learning Enhanced Proactive Retrieval-Augmented Generation). The fundamental objective of ProRAG is to transition dynamic RAG from a reactive paradigm to a proactive one. ProRAG introduces two complementary components, which are outlined below: The first of these is the Prospective Information Needs Prediction (PINP) module, the function of which is to predict future information requirements by means of prospective token generation, thus enabling global retrieval planning. The second is the Core Entity Graph-based Query Expansion (CEQE) module, the purpose of which is to construct and maintain a dynamic entity-relationship graph to generate semantically complete and coherent retrieval queries. Collectively, PINP and CEQE facilitate the proactive identification of necessary information and the subsequent determination of its retrieval method.

The main contributions of this work are threefold:

- **Framework Innovation**: The proposed framework, RL-Enhanced ProRAG, represents a novel integration of proactive retrieval planning with RL-based precision calibration, with the objective of systematically addressing retrieval myopia in multi-hop reasoning.
- **Mechanism Design**: The objective of this study is to design a lightweight RL controller, guided by Cross-Encoder relevance feedback, to dynamically adjust PINP confidence thresholds and CEQE entity filtering parameters. This is intended to enable accurate and efficient proactive retrieval.
- **Empirical Validation**: We conduct extensive experiments on four multi-hop QA benchmarks—2WikiMultihopQA [19], HotpotQA [20], StrategyQA [21], and IIRC [22]—demonstrating that RL-Enhanced ProRAG consistently outperforms state-of-the-art dynamic RAG baselines, with ablation studies confirming the critical role of the RL-based precision control.

II. Related Work

Hallucinations of LLMs. Despite recent advances in instruction following and reasoning [23], [24], large language models still suffer from hallucinations caused by outdated or inaccurate knowledge [25], [26]. Insufficient access to reliable external information often leads to misleading outputs in knowledge-intensive applications.

Retrieval-Augmented Generation. Retrieval-Augmented Generation (RAG) [8], [27] alleviates hallucinations by grounding generation in external documents. While effective in many scenarios, single-round retrieval remains sensitive to retrieval errors, and irrelevant documents may even amplify factual inaccuracies.

Proactive Adaptive Retrieval Strategies. Recent studies have explored multi-turn retrieval as a means of supporting complex reasoning. It is evident that methodologies such as FLARE and DRAGIN initiate retrieval in accordance with local uncertainty signals; however, they predominantly adopt a reactive stance. This propensity engenders retrieval myopia and the generation of incomplete queries. The Proactive Information Need Prediction (PINP) module has been developed to address this limitation by anticipating future information requirements through prospective generation, enabling globally planned retrieval.

Query Formulation and Semantic Enhancement. The retrieval quality is contingent on the semantics of the query. Existing query expansion methods, which are based on static reformulation [28] or pre-constructed knowledge graphs [29], are not capable of capturing dynamic entity relations in multi-hop reasoning. In contrast, the Core Entity Graph-based Query Expansion (CEQE) module utilises a dynamic construction of entity graphs, thereby generating context-aware and semantically complete queries.

Optimization Paradigms for RAG.

The majority of RAG optimisation methods are predicated on the isolation of retrievers or generators. The application of reinforcement learning to enhance robustness or factuality has been demonstrated in the literature [30]. However, it is

important to note that retrieval strategies are typically treated as fixed in such applications. The present work introduces a unified and lightweight RL mechanism that jointly optimises retrieval triggering and query expansion, leveraging cross-encoder relevance signals to bridge retrieval decisions and answer quality. This dimension has received scant attention in the extant literature on adaptive RAG methods, including Self-RAG and SAIL [31].

In summary, extant approaches are deficient in a tightly integrated framework that combines proactive planning, dynamic semantic modelling, and joint optimisation for multi-hop reasoning. RL-Enhanced ProRAG addresses this lacuna by unifying these components in a coherent and efficient retrieval-augmented generation framework.

III. Methods

A. RL-Enhanced ProRAG Framework

1) Overview: This paper proposes the **RL-Enhanced ProRAG (Reinforcement Learning Enhanced Proactive Retrieval-Augmented Generation)** framework, which integrates reinforcement learning with proactive retrieval to address the "retrieval myopia" problem in multi-hop reasoning tasks. As illustrated in Figure 2, the framework comprises three core components:

1) **Proactive Information Need Prediction (PINP) Module**: Anticipates future information requirements through local generation rollouts.
2) **Core Entity Graph-based Query Expansion (CEQE) Module**: Enhances query semantic completeness using dynamic entity relationship graphs.
3) **Reinforcement Learning Unified Controller**: Adaptively calibrates retrieval-related parameters to optimize retrieval decisions.

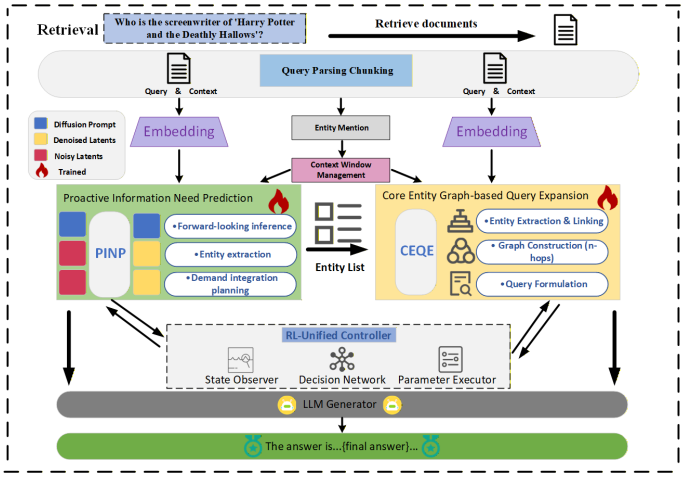


Fig. 2: RL-ProRAG framework overview.

The reinforcement learning controller functions as a high-level coordination mechanism. Rather than directly generating queries or modifying retrieval logic, it dynamically adjusts the internal parameters of the PINP and CEQE modules based on the current reasoning state, enabling adaptive and context-aware retrieval control.

2) Reinforcement Learning Optimization Mechanism:

a) State Representation: At each reasoning step t, the policy observes a six-dimensional state vector:

$$\mathbf{s}_t = \left[\phi_{CE}(q, D_t), \rho_t, \eta_t, \delta_t, \overline{H_t}, \Delta H_t\right]^T \tag{1}$$

where:

- $\phi_{CE}(q, D_t)$ denotes the average relevance score computed by a Cross-Encoder between the query q and the retrieved documents D_t.
- ρ_t represents the confidence score of PINP's prospective information prediction.
- η_t measures the density of the entity graph maintained by the CEQE module.
- δ_t estimates the completion degree of the current reasoning step.
- $\overline{H_t}$ is the average entropy of the model's generation within a sliding window.
- $\Delta H_t = \overline{H_t} - \overline{H_{t-1}}$ captures the temporal trend of uncertainty change.

b) Feature Selection Rationale: The six-dimensional state representation is designed to provide a compact yet comprehensive summary of the multi-hop reasoning context. Specifically, ϕ_{CE} reflects retrieval relevance quality; ρ_t quantifies confidence in anticipated information needs; η_t captures the richness of entity relationships; δ_t indicates reasoning progress; and $\overline{H_t}$ together with ΔH_t monitors generation uncertainty and its dynamics. Ablation experiments show that each feature contributes non-redundantly to performance, with the removal of any single feature resulting in a 1.5–4.2% drop on development data.

c) Action Space: The action vector $\mathbf{a}_t$ consists of three components:

$$\mathbf{a}_t = [a_{\text{retrieve}}, \alpha_p, \alpha_c]^T \tag{2}$$

where:

- $a_{\text{retrieve}} \in \{0, 1\}$ is a binary decision indicating whether to trigger retrieval (1) or continue generation (0).
- $\alpha_p \in \{-0.05, 0, +0.05\}$ adjusts the confidence threshold of the PINP module.
- $\alpha_c \in \{-0.1, 0, +0.1\}$ adjusts the entity filtering threshold used in CEQE.

d) Action Design Rationale: This low-dimensional action space balances expressiveness and stability. The binary variable a_{retrieve} directly controls retrieval triggering, while the discrete adjustment variables α_p and α_c enable fine-grained calibration of PINP and CEQE without introducing training instability.

e) Reward Function: The reward function jointly considers retrieval relevance, reasoning consistency, and efficiency:

$$R(\mathbf{s}_t, \mathbf{a}_t) = 0.7 \cdot R_{CE} + 0.2 \cdot R_{\text{con}} + 0.1 \cdot R_{\text{eff}} \tag{3}$$

We conducted grid search on validation data to determine the weighting coefficients in Eq. 3. Since answer correctness is most strongly correlated with the relevance of retrieved documents, the Cross-Encoder relevance reward R_{CE} receives the highest weight. The reasoning consistency term $R_{consistency}$ provides moderate regularization to prevent topic drift across reasoning steps, while the efficiency term $R_{efficiency}$ penalizes unnecessary retrievals and plays a subsidiary role.
where:

Cross-Encoder Relevance Reward:

$$R_{CE} = \frac{1}{|D_{new}|} \sum_{d \in D_{new}} \phi_{CE}(q, d) \tag{4}$$

Reasoning Consistency Reward:

$$R_{consistency} = \frac{1}{|E_t \cap E_{t-1}|} \sum_{e \in E_t \cap E_{t-1}} \text{sim}_{\text{emb}}(e_t, e_{t-1}) \tag{5}$$

Efficiency Reward:

$$R_{efficiency} = \begin{cases} -0.1 & \text{if } a_{\text{retrieve}} = 1 \text{ and } R_{CE} < 0.5 \\ +0.05 & \text{if } a_{\text{retrieve}} = 0 \text{ and } \overline{H_t} < \theta_{\text{low}} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

f) Policy Network: We employ a lightweight hierarchical policy network:

$$\mathbf{h} = \text{ReLU}(\mathbf{W}_1 \mathbf{s}_t + \mathbf{b}_1) \tag{7}$$

$$\pi_{\text{retrieve}} = \sigma(\mathbf{w}_r^T \mathbf{h} + b_r) \tag{8}$$

$$\pi_{\text{adjust}} = \text{Softmax}(\mathbf{W}_a \mathbf{h} + \mathbf{b}_a) \tag{9}$$

B. PINP Module with RL Enhancement

1) Original PINP Workflow: The PINP module executes three sequential steps:

1) **Prospective Reasoning**: Generate k prospective tokens S_{future}

$$S_{future} = \text{LLM}_{\text{decode}}(P_{\text{prospect}})[:k] \tag{10}$$

2) **Entity Extraction & Disambiguation**: Extract and normalize entities $E^*_{\text{prospective}}$
3) **Need Integration**: Identify new information requirements

$$N_{\text{prospective}} = E^*_{\text{prospective}} \setminus E_{\text{history}} \tag{11}$$

2) RL-Enhanced PINP: With RL enhancement, the need integration step becomes:

$$N^{\text{RL}}_{\text{prospective}} = \{e \in E^*_{\text{prospective}} \mid \text{conf}(e) > \theta_p + \alpha_p\} \setminus E_{\text{history}} \tag{12}$$

where θ_p is the base confidence threshold and α_p is the RL adjustment. This allows dynamic adjustment of entity filtering strictness based on the current reasoning state.

C. CEQE Module with RL Enhancement

Core Entity Graph-based Query Expansion (CEQE) module, augmented with reinforcement learning, is designed to address the *contextual fragmentation* problem in multi-hop reasoning tasks.

Specifically, CEQE operates in three stages: (1) *Entity Extraction and Linking*, where relevant entities are identified from the current context and linked to a knowledge base; (2) *Graph Construction*, where a local subgraph is built by exploring relationships between extracted entities up to n-hops (default $n = 2$); and (3) *Query Formulation*, where the graph structure is used to generate composite queries that explicitly incorporate relational patterns between entities.

The reinforcement learning component further optimizes CEQE's query formulation strategy by learning to balance *query specificity* (focusing on precise entities) and *query coverage* (including relevant contextual relations). Through reward signals based on retrieval relevance and downstream task performance, the RL agent learns to dynamically adjust query expansion parameters, such as the graph exploration depth and entity weight allocation.The specific process is shown in Figure 3.

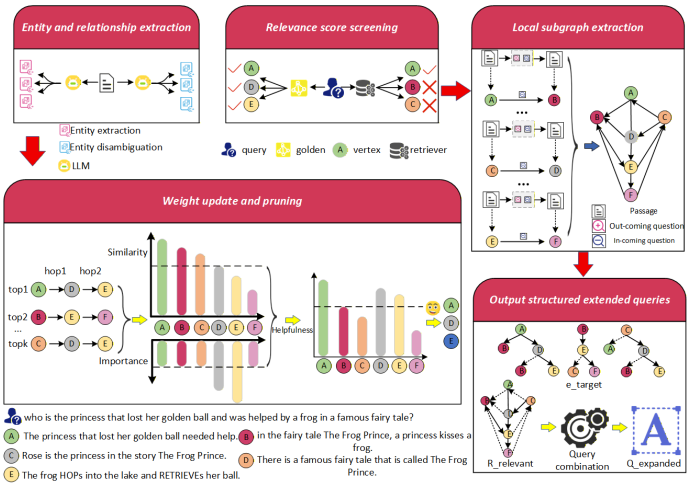


Fig. 3: Workflow of the CEQE module with RL enhancement

1) Original CEQE Workflow: The CEQE module maintains a dynamic entity-relationship graph $G = (V, E)$ and expands queries through:

$$s_{\text{rel}} = \lambda_1 \cdot w_{\text{edge}} + \lambda_2 \cdot \text{Sim}_{\text{text}}(r, q_t) + \lambda_3 \cdot \text{Sim}_{\text{embedding}}(\text{Enc}(e_{\text{neighbor}}), \text{Enc}(q_t)) \tag{13}$$

2) RL-Enhanced CEQE: With RL enhancement, the weights are dynamically adjusted:

$$\lambda_i' = \frac{\lambda_i + \alpha_c \cdot \delta_i}{\sum_{j=1}^{3} (\lambda_j + \alpha_c \cdot \delta_j)} \tag{14}$$

where α_c is the RL adjustment and δ_i are sensitivity coefficients for each dimension. This enables adaptive weighting of different features in entity filtering.

Algorithm 1 RL-Enhanced ProRAG

Require: Query q, knowledge base K, initial parameters θ
Ensure: Answer A

Initialize: Load Cross-Encoder, initialize policy network π_θ, initialize entity graph $G = \emptyset$
$D_{\text{history}} \leftarrow \emptyset$, $E_{\text{history}} \leftarrow \emptyset$
while answer not complete **do**
  Construct state vector $\mathbf{s}_t$ using Equation 1
  Sample action $\mathbf{a}_t \sim \pi_\theta(\cdot|\mathbf{s}_t)$
  if $a_{\text{retrieve}} = 1$ **then**
    Execute RL-calibrated PINP: $N_{\text{prospective}} \leftarrow \text{PINP}_{\text{RL}}(\theta_p + \alpha_p)$
    if $N_{\text{prospective}} \neq \emptyset$ **then**
      Execute RL-calibrated CEQE: $Q_{\text{expanded}} \leftarrow \text{CEQE}_{\text{RL}}(\lambda_i')$
      $D_{\text{new}} \leftarrow \text{Retrieve}(Q_{\text{expanded}}, K)$
      $A_{\text{partial}} \leftarrow \text{LLM\_Generate}(q, D_{\text{history}} \cup D_{\text{new}})$
      Compute reward r_t using Equation 3
      Store $(\mathbf{s}_t, \mathbf{a}_t, r_t)$ in replay buffer
      Update: $D_{\text{history}} \leftarrow D_{\text{history}} \cup D_{\text{new}}$, update entity graph G
    end if
  else
    $A_{\text{partial}} \leftarrow \text{LLM\_Generate}(q, D_{\text{history}})$
  end if
  if replay buffer is full **then**
    Sample batch and update π_θ using PPO
  end if
end while
return Integrate all A_{partial} into final answer A

D. Overall Algorithm

1) Algorithm Pseudocode:

2) Training Strategy: We adopt a three-stage training strategy to ensure stable and efficient optimization:

1) **Imitation Learning**: The policy network is initialized via behavior cloning using an **expert policy** derived from the original ProRAG framework (without RL). This expert triggers retrieval based on a fixed uncertainty threshold ($\overline{H_t} > \tau$) and applies the CEQE module for query expansion. This stage provides a strong and stable initialization for subsequent RL training.
2) **Reinforcement Learning**: The policy is further optimized using Proximal Policy Optimization (PPO) with the reward function defined in Equation 3. During this phase, the agent learns to adaptively calibrate the PINP confidence threshold and CEQE entity filtering parameters according to the reasoning context.
3) **Fine-tuning**: A short domain-specific fine-tuning stage is conducted on target benchmark datasets to improve generalization and robustness.

E. Discussion on Stability and Complexity

1) Convergence and Stability: The reinforcement learning component of RL-Enhanced ProRAG is formulated as a finite-horizon Markov Decision Process (MDP) with bounded state and action spaces. It is posited that, under standard assumptions of bounded rewards and smooth policy updates, the adopted Proximal Policy Optimization (PPO) algorithm ensures stable training and reliable policy improvement in practice.

2) Computational Complexity: We analyze the computational overhead introduced by the RL controller relative to the base ProRAG framework. Let T_{orig} denote the time complexity of the original system, dominated by LLM inference and retrieval operations. The RL-enhanced framework introduces only lightweight additional costs, including state feature computation and policy network inference, both of which incur negligible overhead compared with generation and retrieval.

Moreover, the learned policy reduces redundant retrieval actions, leading to comparable or lower expected retrieval counts than fixed-threshold strategies. As a result, the overall complexity can be expressed as:

$$T_{\text{RL-Enhanced}} \approx T_{\text{orig}} + O(N_{\text{steps}}),$$

where N_{steps} denotes the number of reasoning steps.

3) Comparison with Static Thresholding: Compared to static uncertainty-based retrieval, RL-ProRAG provides three advantages: (1) context-aware decision boundaries, (2) explicit multi-objective optimization via reward function, and (3) adaptive efficiency-accuracy control, leading to more robust retrieval in complex multi-hop reasoning.

IV. Experiment

A. Datasets

The efficacy of the proposed method is evaluated through its application to four representative reasoning benchmarks: 2WikiMultihop QA, HotpotQA, StrategyQA, and IIRC. The 2WikiMultihop QA and HotpotQA are both focused on multi-hop reasoning. IIRC evaluates reading comprehension under incomplete context, while StrategyQA focuses on commonsense reasoning. Collectively, these datasets encompass a wide range of reasoning and knowledge utilisation scenarios.

B. Baselines

We compare ProRAG with representative RAG baselines. Following the experimental setting of FLARE, all multi-round RAG methods are implemented with identical backbone models and retrievers. The only differences lie in retrieval triggering strategies and query formulation mechanisms.

- **wo-RAG**: The LLM answers questions directly without retrieval.
- **SR-RAG**: Single-round retrieval using the original query.
- **FL-RAG**: Retrieval is triggered every n tokens.
- **FS-RAG**: Retrieval is triggered after each sentence.
- **FLARE**: Retrieval is triggered based on generation uncertainty.

C. Selected LLMs

Experiments are conducted using two representative open-source LLMs:

- **LLaMA-2-Chat**: An instruction-tuned LLM optimized for dialogue generation.
- **Vicuna-13B-v1.5**: A conversational model fine-tuned from LLaMA using ShareGPT data.

D. Main Results

Table I present the main experimental results.

We observe several consistent trends:

1) Retrieval augmentation consistently improves performance over wo-RAG, confirming the effectiveness of external knowledge integration.
2) Fixed rule-based multi-round methods (FL-RAG and FS-RAG) do not consistently outperform SR-RAG, indicating that inappropriate retrieval timing may introduce noise and degrade generation quality.
3) RL-ProRAG achieves the best overall performance across most datasets and backbone models, demonstrating the effectiveness of the proposed adaptive retrieval mechanism.
4) Performance gains are more pronounced on multi-hop QA datasets, highlighting the advantages of ProRAG in complex reasoning scenarios.

E. Ablation Study

We conduct ablation studies to analyze the contribution of individual components in ProRAG, including the RL calibrator, the PINP module, and the CEQE module.

Table II reports the ablation results on HotpotQA and 2WikiMultihopQA.

TABLE II: Impact of core modules on multi-hop reasoning tasks.

Model	HotpotQA (EM)	HotpotQA (F1)	2Wiki (EM)	2Wiki (F1)
Full RL-ProRAG	0.330	0.445	0.324	0.418
w/o RL	0.315	0.428	0.310	0.405
w/o PINP	0.290	0.405	0.285	0.380
w/o CEQE	0.305	0.420	0.300	0.395
Standard Dynamic RAG	0.314	0.424	0.304	0.393

As shown in Table II, removing any core component leads to noticeable performance degradation, indicating that PINP, CEQE, and the RL controller are all essential to the overall effectiveness of ProRAG. In particular, disabling PINP causes the largest performance drop on both datasets, highlighting the importance of proactive information need prediction for long-horizon reasoning. Without explicit lookahead planning, the model tends to rely on locally optimal retrieval decisions, resulting in incomplete evidence aggregation.

Removing the CEQE module also leads to consistent performance declines, suggesting that dynamically constructed entity graphs play a crucial role in preserving semantic coherence during multi-hop query formulation. Without structured entity modeling, retrieval queries become more fragmented and less targeted, which negatively affects downstream reasoning accuracy.

When the RL calibrator is removed, the performance decreases more moderately. This observation indicates that the RL component primarily serves as a lightweight coordination mechanism, enabling more effective interaction between PINP and CEQE by adaptively adjusting retrieval thresholds and filtering strategies. Without such calibration, the system relies on static heuristics that are less responsive to contextual variations.

TABLE III: Comparison of retrieval decision mechanisms.

Model	IIRC (EM)	IIRC (F1)	Avg. Ret.
Adaptive Threshold (Full ProRAG)	0.445	0.324	2.8
Fixed Threshold (=0.6)	0.405	0.285	3.5

a) Impact of Adaptive Retrieval Decisions.: To further examine the effectiveness of the proposed RL-based retrieval controller, we compare it with a fixed uncertainty threshold strategy on the IIRC dataset. As shown in Table III, the adaptive mechanism achieves higher EM and F1 scores while reducing the average number of retrieval operations. This improvement suggests that learning-based calibration enables the model to better distinguish between informative and redundant retrieval opportunities, thereby avoiding both premature and excessive evidence acquisition.

TABLE IV: Comparison of query generation strategies.

Model	EM	F1
Graph-Enhanced Query (Full ProRAG)	0.330	0.445
Attention-Based Query (DRAGIN-QFS)	0.314	0.424
Last Sentence Query	0.280	0.395

b) Effectiveness of Query Generation Strategies.: We further evaluate different query formulation methods to assess the contribution of the CEQE module. Table IV reports the performance on HotpotQA. Graph-enhanced queries consistently outperform attention-based and heuristic baselines, indicating that dynamic entity graph construction facilitates more precise and semantically coherent query formulation. By explicitly modeling evolving entity relationships, CEQE reduces semantic drift and improves evidence alignment across multiple reasoning steps.

c) Comparison with Dynamic RAG Frameworks.: We compare RL-ProRAG with representative dynamic RAG systems across multiple benchmarks. As summarized in Table V, our method achieves consistently strong performance across all evaluated datasets, while maintaining competitive retrieval frequency. The results indicate that proactive retrieval planning and reinforcement learning-based calibration jointly contribute to robust evidence acquisition and reasoning accuracy.

Overall, these ablation and comparative studies demonstrate that ProRAG benefits from the tight integration of proactive planning, structured query modeling, and adaptive control. Each component plays a complementary role in mitigating error propagation and context fragmentation in multi-hop reasoning, leading to more reliable and efficient retrieval-augmented generation.

F. Efficiency Analysis

The computational efficiency of ProRAG and baseline methods is evaluated in terms of retrieval frequency, response

TABLE I: Experimental results on Llama2-13b-chat and Vicuna-13b-v1.5

Method	Model	2WikiMultihopQA		HotpotQA		StrategyQA	IIRC	
		EM	F1	EM	F1	Acc.	EM	F1
wo-RAG	Llama2-13b-chat	0.187	0.272	0.223	0.310	0.650	0.168	0.204
SR-RAG	Llama2-13b-chat	0.245	0.336	0.263	0.371	0.654	0.196	0.230
FL-RAG	Llama2-13b-chat	0.217	0.305	0.177	0.268	0.648	0.155	0.186
FS-RAG	Llama2-13b-chat	0.270	0.361	0.267	0.372	0.655	0.171	0.206
FLARE	Llama2-13b-chat	0.224	0.308	0.180	0.276	0.655	0.138	0.167
DRAGIN	Llama2-13b-chat	0.304	0.393	0.314	0.424	0.689	0.185	0.222
RL-ProRAG (ours)	Llama2-13b-chat	0.324	0.418	0.330	0.445	0.695	0.200	0.240
wo-RAG	Vicuna-13b-v1.5	0.146	0.223	0.228	0.326	0.682	0.175	0.215
SR-RAG	Vicuna-13b-v1.5	0.170	0.256	0.254	0.353	0.686	0.217	0.256
FL-RAG	Vicuna-13b-v1.5	0.135	0.213	0.187	0.304	0.645	0.099	0.129
FS-RAG	Vicuna-13b-v1.5	0.188	0.263	0.185	0.322	0.622	0.103	0.134
FLARE	Vicuna-13b-v1.5	0.157	0.226	0.092	0.181	0.599	0.117	0.147
DRAGIN	Vicuna-13b-v1.5	0.252	0.352	0.288	0.416	0.687	0.223	0.265
RL-ProRAG (ours)	Vicuna-13b-v1.5	0.275	0.370	0.305	0.435	0.702	0.240	0.285

TABLE V: Comprehensive comparison with dynamic RAG frameworks.

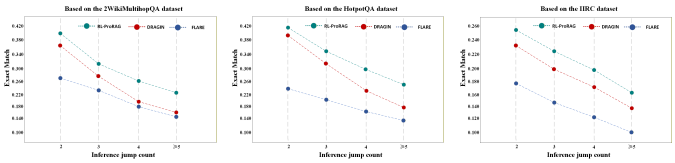
Model	Wiki EM	Hotpot EM	SQA Acc.	IIRC EM	Avg. Ret.
FLARE	0.224	0.180	0.655	0.138	1.6
DRAGIN	0.304	0.314	0.689	0.185	2.8
RL-ProRAG (Ours)	**0.324**	**0.330**	**0.695**	**0.200**	**2.8**

TABLE VI: Efficiency comparison across datasets. Metrics include average retrieval count (#NUM), latency (s), and processed tokens (K).

Model	Metric	2Wiki	Hotpot	SQA	IIRC
FL-RAG	#NUM	3.770	3.194	3.626	3.426
	Lat. (s)	4.23	3.85	3.92	4.15
	Tok. (K)	12.8	11.2	10.5	13.1
FS-RAG	#NUM	3.131	4.583	4.885	4.305
	Lat. (s)	3.65	5.42	5.68	5.13
	Tok. (K)	11.2	15.8	16.2	14.9
FLARE	#NUM	1.592	3.378	0.625	5.521
	Lat. (s)	2.85	4.02	1.95	6.84
	Tok. (K)	9.8	12.5	7.2	18.3
DRAGIN	#NUM	2.631	3.505	4.786	2.829
	Lat. (s)	3.12	4.15	5.52	3.65
	Tok. (K)	10.5	12.8	15.9	11.2
RL-ProRAG	#NUM	2.850	3.400	4.200	2.950
	Lat. (s)	3.05	3.92	4.85	3.42
	Tok. (K)	10.1	12.0	14.2	10.8

latency, and total processed tokens. These factors collectively reflect practical deployment cost.

1) Training and Inference Efficiency: The RL controller is trained using Proximal Policy Optimization (PPO) on 5,000 multi-hop QA instances from HotpotQA and 2WikiMultihopQA. Training converges stably within approximately 30 epochs and requires approximately 12 hours on a single NVIDIA A100 GPU. During the process of inference, the learned policy introduces negligible overhead (less than 5 ms per decision), thus confirming the lightweight nature of the RL module.



(a) 2Wiki (b) Hotpot (c) IIRC

Fig. 4: Complexity analysis across three datasets.

2) Efficiency Trade-offs: As shown in Table VI, RL-ProRAG achieves a balanced trade-off among retrieval frequency, latency, and token consumption across datasets. A comparison of ProRAG with strong baselines such as DRAGIN reveals that ProRAG reduces token usage on HotpotQA (12.0K vs. 12.8K) and latency on IIRC. This finding indicates that ProRAG utilises contexts more efficiently and employs retrieval planning more effectively.

It is evident that, in general, Reinforcement Learning calibration facilitates ProRAG in maintaining a competitive computational cost whilst concomitantly enhancing its reasoning performance.

G. Complexity Sensitivity Analysis

We analyze performance under varying reasoning complexity on HotpotQA, 2WikiMultihopQA, and IIRC. Questions are grouped by hop count.

As shown in Figure 4, ProRAG has been shown to achieve substantially larger gains on high-complexity queries ($\geq$4 hops). On the HotpotQA and 2WikiMultihopQA datasets, relative improvements of 25.0% and 26.2%, respectively, were observed. In the case of 2-hop questions of a simpler nature, the performance gap is reduced.

The findings of this study suggest that the severity of retrieval myopia increases in proportion to the length of reasoning chains. By proactively anticipating future information needs, ProRAG effectively mitigates this limitation in complex reasoning scenarios.

V. Conclusion

This paper introduces ProRAG, a proactive retrieval-augmented generation framework designed to address two critical limitations in dynamic RAG systems for multi-hop reasoning: retrieval myopia and contextual fragmentation. By integrating Proactive Information Need Prediction (PINP) and Core Entity Graph-based Query Expansion (CEQE), ProRAG enables anticipatory retrieval planning and semantically coherent query formulation.

A lightweight reinforcement learning controller is further incorporated to jointly calibrate retrieval timing and query filtering strategies. By leveraging cross-encoder relevance signals and generation uncertainty as dense feedback, the proposed framework adaptively balances retrieval relevance, reasoning consistency, and computational efficiency. Extensive experiments on four multi-hop QA benchmarks demonstrate that RL-Enhanced ProRAG consistently outperforms strong dynamic RAG baselines, while maintaining competitive latency and retrieval frequency.

Despite these encouraging results, several limitations remain. First, the current framework relies on a predefined entity extraction and graph construction pipeline, which may limit its robustness in domains with noisy or implicit entity structures.

Future work will explore more scalable and domain-adaptive entity modelling strategies, as well as more efficient reinforcement learning schemes to reduce training cost. In addition, extending the proposed framework to long-form generation, open-domain dialogue, and tool-augmented reasoning remains a promising direction. We believe that integrating proactive planning, structured knowledge modelling, and RL-driven control provides a solid foundation for advancing adaptive retrieval in large language models.

References

[1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Proc. NeurIPS*, 2022.

[2] OpenAI, "GPT-4 technical report," arXiv:2303.08774, 2023.

[3] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," arXiv:2302.13971, 2023.

[4] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," arXiv:2307.09288, 2023.

[5] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, X. Xia, W. L. Tam, Z. Ma, Y. Xue, J. Zhai, W. Chen, Z. Liu, P. Zhang, Y. Dong, and J. Tang, "GLM-130B: An open bilingual pre-trained model," in *Proc. ICLR*, 2023.

[6] Z. Du, Y. Qian, X. Liu, M. Ding, J. Qiu, Z. Yang, and J. Tang, "GLM: General language model pretraining with autoregressive blank infilling," in *Proc. ACL*, 2022, pp. 320–335.

[7] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, pp. 248:1–248:38, 2023.

[8] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-augmented language model pre-training," arXiv:2002.08909, 2020.

[9] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie, and J. Li, "A survey of knowledge enhanced pre-trained language models," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 4, pp. 1413–1430, 2024.

[10] H. Li, Y. Su, D. Cai, Y. Wang, and L. Liu, "A survey on retrieval-augmented text generation," arXiv:2202.01110, 2022.

[11] C.-H. Tan, J.-C. Gu, C. Tao, Z.-H. Ling, C. Xu, H. Hu, X. Geng, and D. Jiang, "TegTok: Augmenting text generation via task-specific and open-world knowledge," in *Findings of ACL*, 2022, pp. 1597–1609.

[12] Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig, "Active retrieval augmented generation," in *Proc. EMNLP*, 2023, pp. 7969–7992.

[13] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," arXiv:2310.11511, 2023.

[14] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, and J. Larson, "From local to global: A graph RAG approach to query-focused summarization," arXiv:2404.16130, 2024.

[15] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis, "Generalization through memorization: Nearest neighbor language models," arXiv:1911.00172, 2019.

[16] S. Borgeaud et al., "Improving language models by retrieving from trillions of tokens," arXiv:2112.04426, 2021.

[17] Q. Liu, C. Zhang, C. Zheng, G. Hu, X. Li, and Z. Zhang, "Beyond the answer: Advancing multi-hop QA with fine-grained graph reasoning and evaluation," in *Proc. ACL*, 2025, pp. 23433–23456.

[18] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, Q. Guo, M. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," arXiv:2312.10997, 2023.

[19] X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa, "Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps," arXiv:2011.01060, 2020.

[20] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "HotpotQA: A dataset for diverse, explainable multi-hop question answering," in *Proc. EMNLP*, 2018, pp. 2369–2380.

[21] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant, "Did Aristotle use a laptop? A question answering benchmark with implicit reasoning strategies," *TACL*, vol. 9, pp. 346–361, 2021.

[22] J. Ferguson, M. Gardner, H. Hajishirzi, T. Khot, and P. Dasigi, "IIRC: A dataset of incomplete information reading comprehension questions," in *Proc. EMNLP*, 2020, pp. 1137–1147.

[23] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, Q. V. Do, Y. Xu, and P. Fung, "A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity," in *Proc. IJCNLP*, 2023, pp. 675–718.

[24] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang, "Is ChatGPT a general-purpose natural language processing task solver?" in *Proc. EMNLP*, 2023, pp. 1339–1384.

[25] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston, "Retrieval augmentation reduces hallucination in conversation," in *Findings of EMNLP*, 2021, pp. 3784–3803.

[26] S. M. T. I. Tonmoy, S. M. M. Zaman, V. Jain, A. Rani, V. Rawte, A. Chadha, and A. Das, "A comprehensive survey of hallucination mitigation techniques in large language models," arXiv:2401.01313, 2024.

[27] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. NeurIPS*, 2020.

[28] J. Kim, J. Nam, S. Mo, J. Park, S.-W. Lee, M. Seo, J.-W. Ha, and J. Shin, "SuRe: Summarizing retrievals using answer candidates for open-domain QA of LLMs," in *Proc. ICLR*, 2024.

[29] T. Zhang, S. G. Patil, N. Jain, S. Shen, M. Zaharia, I. Stoica, and J. E. Gonzalez, "RAFT: Adapting language model to domain specific RAG," arXiv:2403.10131, 2024.

[30] O. Yoran, T. Wolfson, O. Ram, and J. Berant, "Making retrieval-augmented language models robust to irrelevant context," in *Proc. ICLR*, 2024.

[31] H. Luo, Y.-S. Chuang, Y. Gong, T. Zhang, Y. Kim, X. Wu, D. Fox, H. Meng, and J. R. Glass, "SAIL: Search-augmented instruction learning," arXiv:2305.15225, 2023.