ACT-ON-GRAPH: RECOVERABILITY-AWARE MULTI-HOP RETRIEVAL FOR RELIABLE KNOWLEDGE GRAPH-GROUNDED LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) exhibit strong reasoning capabilities, but they remain prone to hallucinations due to outdated parametric knowledge. Knowledge graphs (KGs) provide structured and interpretable evidence for factual grounding, making reliable retrieval essential for KG-enhanced LLM reasoning. However, existing methods typically focus on local expansion or complete-path ranking, with limited attention to whether intermediate retrieval states can still lead to a correct answer. Consequently, early retrieval mistakes may cause cascading errors, while excessive graph expansion introduces irrelevant candidate paths. We propose Act-on-Graph (AoG), a recoverability-aware multi-hop retrieval framework that formulates retrieval as a sequential decision process. Rather than evaluating only immediate actions or completed paths, AoG explicitly models the recoverability of intermediate retrieval states. Specifically, AoG converts answer-reaching paths into step-wise state–action and prefix-state supervision, jointly learning a local action policy for selecting relation–entity actions and a prefix-state value function for estimating state recoverability. During inference, an uncertainty-aware beam search retains plausible alternatives under ambiguous decisions while pruning redundant and low-recoverability branches. Experiments on WebQSP and CWQ show that AoG improves Hits@1 by 1.3% and 1.2% over the strongest baseline, respectively.

1 INTRODUCTION

Large language models (LLMs) (Achiam et al., 2023; Touvron et al., 2023; Guo et al., 2025) have demonstrated strong reasoning capabilities across a wide range of tasks, particularly question answering (Brown et al., 2020; Chowdhery et al., 2023; Li et al., 2024). However, their parametric knowledge is costly to update and may be incomplete or outdated, leading to factual errors and unfaithful reasoning (Luo et al., 2025; Chen et al., 2025). Retrieval-augmented generation (RAG) (Lewis et al., 2020) mitigates this limitation by grounding LLMs in external knowledge, while knowledge graphs (KGs) provide structured and interpretable evidence. Consequently, reliable multi-hop evidence retrieval becomes a key bottleneck in KG-enhanced reasoning.

Existing KG retrieval approaches mainly follow two paradigms: (1) agent-based methods (Sun et al., 2024; Xu et al., 2024), which use LLMs to iteratively interact with the KG and explore candidate paths, and (2) subgraph-based methods (Luo et al., 2024; Li et al., 2025a; Lin et al., 2026), which retrieve paths or subgraphs before downstream reasoning. Despite recent progress, two challenges remain. First, many methods primarily rank local expansions or complete paths without explicitly evaluating whether an intermediate state can still reach an answer within the remaining hop budget. Importantly, deviation from a currently preferred or answer-reaching path does not necessarily imply retrieval failure, since the resulting state may still reach the answer through an alternative continuation. Conversely, an apparently relevant local decision may lead the search into an unrecoverable state, causing ***cascading error propagation***. Second, ***path explosion*** causes the candidate space to grow combinatorially with retrieval depth, introducing irrelevant branches and increasing retrieval cost.

To better understand these retrieval failures, we conduct a fine-grained failure analysis of representative baselines on WebQSP and CWQ. For each incorrectly answered question, we compare retrieved paths with answer-reaching paths obtained by bounded graph search under the same hop constraint.

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.

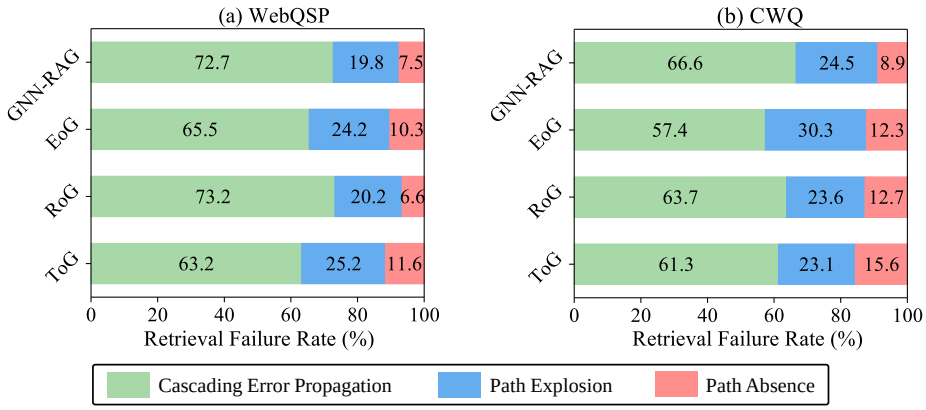


Figure 1: Distribution of retrieval failure types across baselines on WebQSP and CWQ.

We categorize retrieval failures into three types. **(1) Cascading error propagation**, where an early deviation makes the gold answer unreachable within the remaining budget; **(2) Path explosion**, where irrelevant branches overwhelm the search and prune correct continuations; and **(3) Path absence**, where the correct path does not exist in the knowledge graph. As shown in Figure 1, cascading error propagation and path explosion account for the majority of failures, indicating that most errors originate from ineffective retrieval decisions during the search process rather than from the absence of valid knowledge graph paths.

These observations suggest that reliable multi-hop retrieval requires more than accurate local ranking. In particular, a retriever should distinguish two complementary questions: whether a candidate action is semantically relevant to the current question and reasoning history, and whether the resulting state remains structurally capable of reaching an answer within the remaining hop budget. Existing methods rarely model these two signals explicitly within a unified retrieval process. This motivates **Act-on-Graph** (**AoG**), a recoverability-aware framework for multi-hop knowledge graph retrieval.

AoG formulates multi-hop retrieval as a sequential decision process and explicitly decouples local action relevance from prefix-state recoverability. It transforms answer-reaching paths into step-wise state–action supervision and prefix-state recoverability supervision. A question- and history-conditioned action policy estimates the semantic relevance of candidate transitions, while a task-specific value function evaluates whether the resulting state can still reach a correct answer within the remaining hop budget. Importantly, recoverability reflects structural answer reachability rather than the immediate correctness of the current prefix, capturing whether retrieval can still recover from the current state. These two signals are therefore complementary. During inference, AoG integrates them through an uncertainty-aware beam search that preserves plausible alternatives under ambiguous decisions while pruning redundant or low-recoverability branches. The highest-scoring paths are then aggregated as structured KG evidence for downstream LLM reasoning.

Our main contributions are threefold: **(1)** We formulate multi-hop KG retrieval as a recoverability-aware sequential decision process that explicitly separates local action relevance from prefix-state recoverability under the remaining hop budget, providing a way to mitigate cascading retrieval errors. **(2)** We propose AoG, which learns complementary action-policy and prefix-value signals from step-wise supervision and integrates them into an uncertainty-aware search procedure for reliable multi-hop retrieval while reducing redundant exploration. **(3)** We conduct extensive experiments across multiple KGQA benchmarks, showing that AoG retrieves more reliable multi-hop evidence paths and improves downstream LLM reasoning with efficient graph exploration.

2 Related Work

LLM Reasoning. Numerous studies have explored prompting-based strategies to enhance the reasoning capabilities of LLMs. Representative approaches include Chain-of-Thought (CoT) (Wei et al., 2022) and its variants, such as Auto-CoT (Zhang et al., 2023), CoTKR (Wu et al., 2024), and Pattern-CoT (Zhang et al., 2025). Beyond linear reasoning chains, ToT (Yao et al., 2023) and

GoT (Besta et al., 2024) further enable the structured exploration of multiple reasoning trajectories. However, these methods still rely heavily on the parametric knowledge of LLMs. Retrieval-augmented generation (RAG) alleviates this limitation by incorporating external evidence into the reasoning process. Early approaches (Izacard et al., 2023; Fan et al., 2024; Cai et al., 2025) mainly adopt single-step retrieval, whereas iterative RAG methods (Asai et al., 2024; Yang et al., 2024b; Guan et al., 2026) perform multi-step retrieval and reasoning to progressively gather relevant evidence. Nevertheless, their reliance on unstructured text limits explicit modeling of multi-hop dependencies and construction of interpretable reasoning paths, motivating the use of structured knowledge graphs.

Knowledge Graph-Enhanced LLM Reasoning. KGs provide structured knowledge for LLM reasoning. Earlier studies integrate KG knowledge through pre-training (Zhang et al., 2022b) or fine-tuning (Chen et al., 2025), but such approaches are costly and difficult to update (Pan et al., 2024). Therefore, recent works have explored leveraging KGs as external knowledge during inference (Chen et al., 2024; Edge et al., 2024). Among them, agent-based methods, such as ToG (Sun et al., 2024) and GoG (Xu et al., 2024), iteratively navigate relations and entities but require frequent LLM interactions. Subgraph-based methods, including RoG (Luo et al., 2024), GNN-RAG (Mavromatis & Karypis, 2025), ReKnoS (Wang et al., 2025), EoG (Li et al., 2025b), ProgRAG (Park et al., 2026), and SCR (Lin et al., 2026) retrieve candidate paths more efficiently, but they generally do not explicitly distinguish local semantic action relevance from remaining-budget structural recoverability.

3 METHODOLOGY

Figure 2 shows the architecture of AoG, which consists of four components: (1) step-wise supervision construction, which converts answer-reaching paths into state–action and prefix-state instances; (2) local action policy learning, which predicts question- and history-aware relation–entity actions; (3) prefix-state value learning, which estimates whether a partial path can reach a correct answer within the remaining hop budget; and (4) uncertainty-aware inference, which combines the policy and value function to guide beam expansion, evidence aggregation, and final LLM reasoning.

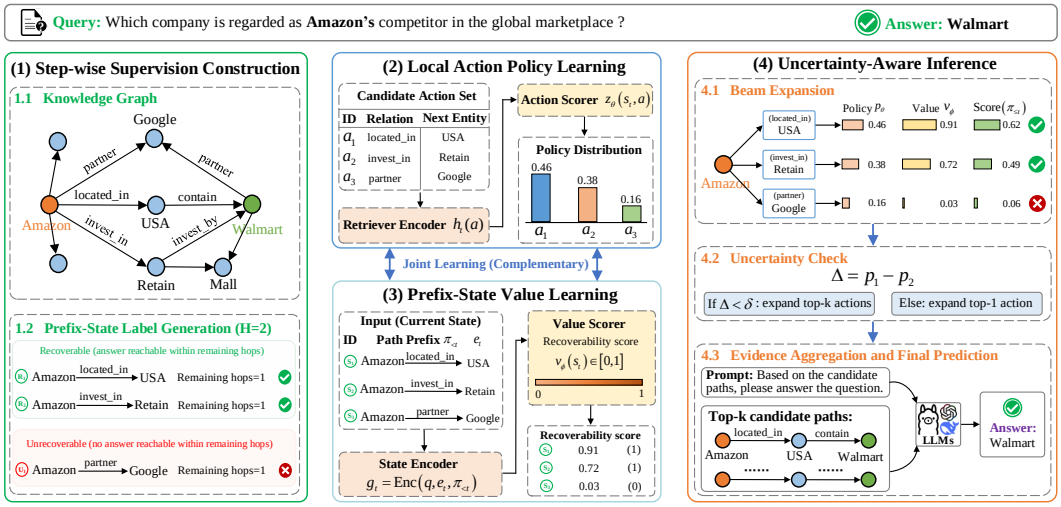


Figure 2: The overall framework of **AoG**. (1) Step-wise supervision converts answer-reaching paths into action labels and prefix recoverability labels. (2) Local action policy learning predicts history-aware relation–entity actions for path expansion. (3) Prefix-state value learning estimates whether a partial path can still reach the correct answer within the remaining hop budget. (4) Uncertainty-aware inference adaptively expands and aggregates candidate paths for final LLM prediction.

3.1 PROBLEM FORMULATION

Given a question q, a knowledge graph G, topic entities $\mathcal{E}q$, and a maximum number of hops H, AoG retrieves multi-hop candidate paths starting from $e_0 \in \mathcal{E}q$ and selects the top-k paths as structured

evidence for LLM reasoning. We formulate multi-hop retrieval as a sequential decision process. At step t, the retrieval state is defined as $s_t = (q, e_t, \pi_{<t})$, where e_t denotes the current entity and $\pi_{<t}$ denotes the reasoning prefix before step t. The available action space consists of all outgoing relation–entity transitions from e_t:

$$\mathcal{A}(s_t) = \{(r, e') \mid (e_t, r, e') \in G\} \tag{1}$$

Executing $a_t = (r_t, e_{t+1})$ extends the prefix and transitions to s_{t+1}. AoG jointly models two complementary signals for the decision process: an action policy $p_\theta(a_t \mid s_t)$ that measures the relevance of next-hop actions, and a state value function $v_\phi(s_t)$ that assesses whether the current prefix remains recoverable within the remaining hop budget. The former guides fine-grained local path expansion, while the latter discourages the search from continuing along unrecoverable prefixes.

3.2 Step-wise Supervision Construction

In KGQA, supervision is typically provided as complete answer-reaching paths, whereas retrieval proceeds step by step. We therefore decompose each answer-reaching path into state–action samples and prefix-state recoverability samples.

For each question q, let Π_q^* denote the set of answer-reaching paths within the hop limit H. Each path $\pi^* \in \Pi_q^*$ is represented as $\pi^* = [(e_t, r_t, e_{t+1})]_{t=0}^{L-1}$, where $L \le H$. Since multiple answer-reaching paths may exist, a prefix state can have multiple positive next actions. For $s_t = (q, e_t, \pi_{<t})$, we define

$$\mathcal{A}^+(s_t) = \{(r_t^*, e_{t+1}^*) \mid \pi^* \in \Pi_q^*,\ \pi_{<t}^* = \pi_{<t}\} \tag{2}$$

where (r_t^*, e_{t+1}^*) is the t-th relation–entity action of π^*. In this way, AoG preserves multiple answer-reaching continuations from the same prefix rather than forcing the model to imitate a single selected path.

For value learning, let $\mathcal{Y}_q$ be the set of gold answer entities. A state is recoverable if its current entity can still reach any gold answer within the remaining hop budget:

$$y_t^{\mathrm{val}} = \mathbb{I}\Big[\exists y \in \mathcal{Y}_q :\ e_t \xrightarrow{\le H-t} y\Big] \tag{3}$$

Here, recoverability captures structural answer reachability rather than semantic correctness in isolation. Semantic relevance to the question and reasoning history is provided by the question- and history-conditioned action policy, making the two signals complementary: the policy evaluates local action relevance, while the value function estimates whether the resulting state remains answer-reachable within the remaining hop budget.

Recoverability labels are computed offline by bounded graph search using only training-set gold answers, introducing no additional inference-time graph-search cost. At inference, recoverability is estimated solely by the learned value function. To construct harder training samples, we perturb answer-reaching paths with non-gold actions and re-label the resulting prefixes according to Eq. 3, since a perturbed prefix may still reach a correct answer through an alternative path.

3.3 Local Action Policy Learning

For each state–action pair (s_t, a) with $a = (r, e')$, we serialize the question, current entity, reasoning prefix, and candidate action into a textual input $x_t(a) = \mathrm{Format}(q, e_t, \pi_{<t}, a)$. The retriever backbone encodes this serialized input and produces a contextual representation $h_t(a)$ for the state–action pair. The action score is then computed as $z_\theta(s_t, a) = \mathbf{w}_{\mathrm{act}}^\top h_t(a) + b_{\mathrm{act}}$. The local action policy is defined by a softmax over all candidate actions under the same state:

$$p_\theta(a \mid s_t) = \frac{\exp(z_\theta(s_t, a))}{\sum_{a' \in \mathcal{A}(s_t)} \exp(z_\theta(s_t, a')} \tag{4}$$

In addition to gold-action supervision, we further introduce a path-prior distillation signal to encourage the policy to capture relation-level guidance from answer-reaching paths. Let $\mathcal{R}^+(s_t)$ denote the set of relations appearing in the gold actions $\mathcal{A}^+(s_t)$. We assign a higher teacher score to candidate actions whose relations match a gold relation:

$$\tilde{z}_t(a) = \begin{cases} 1, & r(a) \in \mathcal{R}^+(s_t) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $r(a)$ denotes the relation of action a. The teacher distribution is then computed as:

$$\tilde{p}_t(a) = \frac{\exp(\tilde{z}_t(a)/\tau)}{\sum_{a' \in \mathcal{A}(s_t)} \exp(\tilde{z}_t(a')/\tau)} \tag{6}$$

The action policy is trained with the following loss:

$$\mathcal{L}_{\text{act}} = -\log \sum_{a \in \mathcal{A}^+(s_t)} p_\theta(a \mid s_t) + \lambda_{\text{kd}} \text{KL}(\tilde{p}_t(\cdot) \| p_\theta(\cdot \mid s_t)) \tag{7}$$

where λ_{kd} controls the weight of the distillation term.

3.4 Prefix-State Value Learning

Local action relevance alone cannot fully prevent cascading errors, because a semantically plausible action may still lead to a structurally unrecoverable state. Therefore, AoG learns a prefix-state value function to estimate whether the resulting state remains answer-reachable within the remaining hop budget. For each prefix state, the retriever backbone maps the question, current entity, and reasoning prefix into a state representation $g_t = \text{Enc}(q, e_t, \pi_{<t})$. The value estimation head then predicts the recoverability score as $v_\phi(s_t) = \sigma(\mathbf{w}_{\text{val}}^\top g_t + b_{\text{val}})$. Here, $v_\phi(s_t) \in (0, 1)$ estimates the probability that the current prefix can still reach a correct answer within the remaining hop budget. The value function is trained with binary cross-entropy loss:

$$\mathcal{L}_{\text{val}} = -y_t^{\text{val}} \log v_\phi(s_t) - (1 - y_t^{\text{val}}) \log(1 - v_\phi(s_t)) \tag{8}$$

The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \lambda_{\text{val}} \mathcal{L}_{\text{val}} \tag{9}$$

where λ_{val} controls the trade-off between action policy learning and value function learning.

3.5 Uncertainty-Aware Inference

After training, AoG performs multi-hop retrieval using the learned action policy and prefix-state value function. The inference process contains two steps: uncertainty-aware beam search and evidence path aggregation.

3.5.1 Uncertainty-Aware Beam Search

Starting from the topic entities, AoG conducts beam search for at most H steps. The overall process is illustrated in Figure 3 and the complete retrieval procedure is provided in Appendix A. At each state s_t, the action policy $p_\theta(a \mid s_t)$ assigns scores to all candidate actions. For an expanded prefix $\pi_{\le t}$ generated by the action sequence $\{a_i\}_{i=0}^t$, AoG computes its retrieval score as follows:

$$\text{score}(\pi_{\le t}) = \frac{1}{t+1} \sum_{i=0}^{t} \log p_\theta(a_i \mid s_i) + \alpha \log v_\phi(s_{t+1}) \tag{10}$$

where α controls the contribution of the value function. The first term measures the average local action confidence, while the second term evaluates the recoverability of the resulting state.

To avoid premature pruning under ambiguous decisions, AoG adaptively adjusts the number of expanded actions according to policy uncertainty. Let p_1 and p_2 denote the largest and second-largest action probabilities at state s_t, respectively. If $p_1 - p_2 < \delta$, the current decision is regarded as uncertain, and AoG allows multiple top-ranked actions to enter the candidate prefix set under the global beam budget w. Otherwise, only the top-ranked action is expanded.

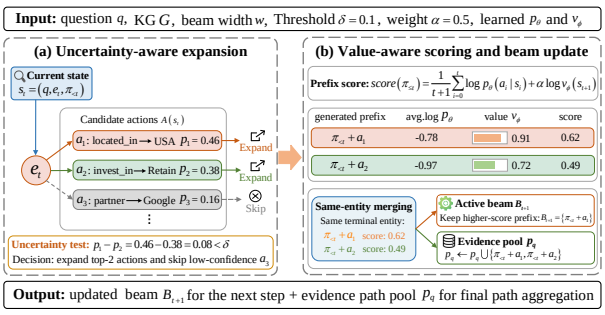


Figure 3: Illustration of the uncertainty-aware beam search.

After expansion, all generated prefixes are scored by Eq. 10 and added to an evidence pool for final path selection. To reduce redundant exploration, AoG applies same-entity merging only to the active beam used for subsequent expansion. Specifically, when multiple prefixes terminate at the same entity, only the highest-scoring prefix is retained for further expansion as an efficiency-oriented approximation. Since such prefixes may encode different reasoning histories, the remaining prefixes are not discarded but preserved in the evidence pool as candidate evidence. Finally, at most w prefixes are retained in the active beam for the next step.

3.5.2 Evidence Path Aggregation and LLM Reasoning

After beam search, AoG selects the top-k paths according to their final retrieval scores and uses them as structured evidence. Each selected path Π is verbalized into a textual reasoning chain $C(\Pi) = \mathrm{Verbalize}(\Pi)$. The evidence context is constructed by concatenating the verbalized paths $C(q) = \mathrm{Concat}\big(C(\Pi_1), C(\Pi_2), \ldots, C(\Pi_k)\big)$.

Given the question q and the evidence context $C(q)$, the LLM predicts the final answer as follows:

$$\hat{y} = \mathrm{LLM}\big(\mathrm{Prompt}(q, C(q))\big) \tag{11}$$

Compared to relying on a single path, AoG aggregates multiple high-scoring paths to provide richer evidence and reduce the risk of depending on incomplete or erroneous single reasoning chains.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate AoG on two widely used multi-hop KGQA benchmarks, WebQSP (Yih et al., 2016) and CWQ (Talmor & Berant, 2018), both built on Freebase (Bollacker et al., 2008). WebQSP mainly contains 1-2 hop questions, while CWQ involves more challenging compositional reasoning with longer reasoning chains. Detailed dataset statistics are provided in Appendix B.1.

Evaluation Metrics. Following prior work (Um et al., 2026; Li et al., 2025a), we report Hits@1 and F1 scores to evaluate downstream KGQA performance. For retrieval evaluation, we use Hit@k and Path Recall@k, where Path Recall@k measures whether the top-k retrieved paths include at least one complete path connecting a topic entity to the corresponding gold answer entity. Detailed metric definitions are provided in Appendix B.2.

Baselines. We compare AoG with 19 baselines grouped into four categories: 1) Embedding-based methods, 2) Retrieval-based methods, 3) LLMs, and 4) LLMs+KGs methods. Unless otherwise specified, the results in Table 1 are taken from the corresponding papers under their reported experimental settings. The details of each baseline are described in Appendix B.3.

Implementation. In our implementation, we use a fine-tuned Llama3.1-8B as the AoG retriever backbone and GPT-4o-mini as the final reasoning model. The retriever is fine-tuned with AdamW for 4 epochs using a batch size of 8 and a learning rate of 2×10^{-4}. We adopt a cosine scheduler with a warmup ratio of 0.03. The uncertainty threshold δ and recoverability weight α are set to 0.1 and 0.5, respectively. During inference, AoG performs uncertainty-aware beam search over the KG, with beam sizes $w = 8$ for WebQSP and $w = 12$ for CWQ. We retain the top-k candidate paths for evidence aggregation, where $k = 25$ for both datasets. Parameter sensitivity analysis is provided in Appendix C.2. Our model is implemented in PyTorch and trained on 4 NVIDIA A100 GPUs.

4.2 Main Results Comparison

In this section, we compare AoG with representative baselines on KGQA tasks. The results are reported in Table 1. AoG achieves the best Hits@1 on both datasets and the best F1 on WebQSP, while obtaining the second-best F1 on CWQ. Specifically, on WebQSP, AoG improves Hits@1 by 1.3% and F1 by 1.4%, respectively, over the second-best method SCR. On the more challenging CWQ dataset, AoG achieves improvements of 11.9% and 7.5% in Hits@1 and F1, respectively, compared with the strong baseline RoG. Although AoG slightly lags behind EoG in F1 on CWQ, it achieves a higher

Hits@1 score. This may be attributed to AoG's prioritization of high-confidence and recoverable paths for accurate top-1 prediction, whereas EoG focuses more on broad candidate path coverage to improve reasoning recall in complex multi-answer scenarios. Among different categories, retrieval-based methods generally outperform embedding-based methods, indicating that explicitly retrieving relevant subgraphs or candidate paths is beneficial for KGQA. In contrast, although LLM-based methods possess strong general reasoning capabilities, their performance remains limited without explicit KG evidence. LLMs+KGs methods achieve the most competitive results, with methods such as RoG, EoG, SCR, and AoG showing clear advantages over other baselines. Overall, these results demonstrate that recoverability-aware path retrieval enables AoG to retrieve reliable evidence paths, thereby improving the reliability of KG-enhanced LLM reasoning.

Table 1: Performance comparison (%) on two KGQA datasets. The best and second-best results within each category are highlighted in **bold** and underlined, respectively.

Types	Methods	WebQSP		CWQ	
		Hits@1	F1	Hits@1	F1
Embedding	KV-Mem (Miller et al., 2016)	46.7	34.5	18.4	15.7
	NSM (He et al., 2021)	68.7	62.8	47.6	42.4
	TransferNet (Shi et al., 2021)	71.4	–	48.6	–
	KGT5 (Saxena et al., 2022)	56.1	–	36.5	–
Retrieval	GraftNet (Sun et al., 2018)	66.4	60.4	36.8	32.7
	SR+NSM (Zhang et al., 2022a)	68.9	64.1	50.2	47.1
	SR+NSM+E2E (Zhang et al., 2022a)	69.5	64.1	49.3	46.3
	ReaRev (Mavromatis & Karypis, 2022)	76.4	70.9	52.9	47.8
LLMs	Qwen2-7B (Yang et al., 2024a)	50.8	35.5	25.3	21.6
	Llama2-7B (Touvron et al., 2023)	56.4	36.5	28.4	21.4
	Llama3.1-8B (Grattafiori et al., 2024)	55.5	34.8	28.1	22.4
	GPT-4o-mini (Achiam et al., 2023)	61.8	43.6	38.2	32.9
LLMs+KGs	ToG (Sun et al., 2024)	82.6	–	67.6	–
	RoG (Luo et al., 2024)	85.7	70.8	62.6	56.2
	GNN-RAG (Mavromatis & Karypis, 2025)	85.7	71.3	66.8	59.4
	ReKnoS (Wang et al., 2025)	83.8	–	66.8	–
	EoG (Li et al., 2025b)	85.0	74.1	70.8	**65.1**
	ProgRAG (Park et al., 2026)	90.4	–	73.3	–
	SCR (Lin et al., 2026)	90.8	74.8	68.4	57.3
	AoG	**92.1**	**76.2**	**74.5**	63.7

4.3 Analysis of Retrieval Effectiveness

4.3.1 Effect of Retrieval Quality on KGQA Performance

Table 2 compares different retrieval methods on CWQ in terms of both retrieval quality and downstream reasoning performance. AoG achieves the best overall retrieval quality while using the most compact KG context. Compared with SCR, AoG improves Path Recall@10 from 65.9 to 72.7, demonstrating that the proposed recoverability-aware search is more effective at preserving complete answer-reaching paths, rather than merely retrieving answer-related entities. This result directly supports our motivation that effective multi-hop retrieval should consider not only

Table 2: Comparison of different methods on CWQ. #KG Tokens denote the average number of KG tokens retrieved as context for the LLM.

Method	Retrieval				Reasoning
	#KG Tokens	Hit@1	Hit@10	Path Recall@10	Hits@1
ToG	4804	46.5	70.7	61.2	67.6
RoG	635	28.8	66.7	54.7	62.6
GNN-RAG	518	45.2	70.1	60.0	66.8
SCR	263	53.4	72.3	65.9	68.4
AoG	**222**	**58.8**	**76.1**	**72.7**	**74.5**

local action relevance but also the recoverability of intermediate states. Moreover, AoG uses only 222 tokens on average, fewer than all compared baselines, indicating that it can reduce redundant graph exploration and provide concise yet informative candidate paths for the LLM. Benefiting from more complete candidate paths and less noisy context, AoG achieves the best downstream Hits@1 on CWQ. These results suggest that prefix-state recoverability helps preserve complete answer-reaching paths while reducing redundant graph context. The results on WebQSP are reported in Appendix C.3.

4.3.2 Fine-grained Retrieval Failure Reduction

To further investigate whether AoG alleviates cascading error propagation and path explosion during multi-hop retrieval, we conduct a fine-grained failure analysis by comparing retrieved paths with gold answer-reaching paths for answerable test questions. As shown in Figure 4, AoG substantially reduces both types of retrieval failures compared with strong baselines, indicating that the prefix-state value function helps avoid unrecoverable intermediate states, while uncertainty-aware beam search preserves plausible continuations under ambiguous decisions. Together with the retrieval quality results in Table 2, these findings indicate that AoG is capable of retrieving more complete and reliable multi-hop paths.

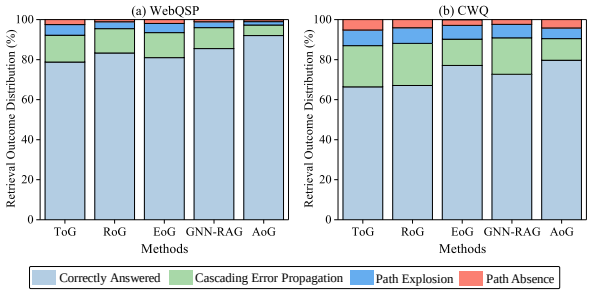


Figure 4: Fine-grained retrieval outcome distribution.

4.4 Ablation Study

To evaluate the contribution of each component in AoG, we consider six variants: (1) **w/o prefix-state value**, which removes the learned recoverability signal; (2) **w/o policy score**, which removes the policy term from prefix scoring; (3) **w/o path-prior distillation**, which removes the relation-level distillation objective by setting $\lambda_{\text{kd}} = 0$; (4) **w/o uncertainty-aware expansion**, which replaces adaptive expansion with standard fixed beam search; (5) **w/o same-entity merging**, which disables prefix merging for paths terminating at the same entity; and (6) **w/ random expansion**, which replaces the learned action policy with random action selection. As shown in Table 3, removing either the policy score or the prefix-state value causes the largest performance degradation, confirming that semantic action relevance and structural recoverability provide complementary signals for multi-hop retrieval. Uncertainty-aware expansion and same-entity merging also consistently improve performance by preserving plausible alternatives and reducing redundant exploration, respectively. Removing path-prior distillation also results in a slight performance degradation, indicating that the relation-level distillation objective provides additional guidance for learning more effective action selection patterns. Random expansion performs substantially worse than the full model, further validating the importance of policy-guided retrieval. Overall, these results support the joint design of semantic action relevance, prefix-state recoverability, and uncertainty-aware graph exploration.

Table 3: Ablation study of AoG core modules.

Method	WebQSP		CWQ	
	Hits@1	F1	Hits@1	F1
AoG	**92.1**	**76.2**	**74.5**	**63.7**
w/o prefix-state value	84.7	69.2	62.1	56.2
w/o policy score	84.1	68.7	61.5	56.7
w/o path-prior distillation	89.9	73.3	71.8	61.6
w/o uncertainty-aware expansion	90.6	74.7	72.5	61.6
w/o same-entity merging	88.5	75.1	69.6	62.2
w/ random expansion	63.6	54.6	45.2	43.3

4.5 Performance Comparison with Different LLMs

To provide a controlled comparison across different backbones, we re-evaluate all KG-enhanced LLM methods reported in Table 4 using the same reasoning LLM for each backbone setting while keeping the remaining evaluation protocol fixed. Table 4 reports the Hits@1 results on WebQSP and CWQ. Across the three reasoning backbones, AoG consistently achieves competitive performance, showing strong adaptability to LLMs of different scales. The results also show that AoG benefits

from stronger reasoning models. AoG's performance improves steadily with stronger reasoning backbones, from Llama2-7B to Llama3.1-8B and GPT-4o-mini. Overall, these results demonstrate that the effectiveness of AoG is not tied to a particular reasoning backbone and that stronger reasoning models can further benefit from high-quality retrieved evidence.

Table 4: Comparison of Hits@1 (%) under different LLM backbones on WebQSP and CWQ.

Method	WebQSP			CWQ		
	Llama2-7B	Llama3.1-8B	GPT-4o-mini	Llama2-7B	Llama3.1-8B	GPT-4o-mini
RoG	85.7	85.2	86.2	62.6	61.7	64.3
GNN-RAG	85.7	86.0	87.1	66.8	67.1	67.8
ReKnoS	64.7	67.9	83.8	51.2	56.7	66.8
EoG	81.1	81.6	85.0	67.3	**70.9**	70.8
ProgRAG	85.3	87.1	90.4	67.2	69.6	73.3
SCR	86.5	88.7	90.8	65.1	67.3	68.4
AoG	**87.2**	**89.5**	**92.1**	**67.5**	70.4	**74.5**

4.6 Case Study

We present a qualitative case study in Table 5 to illustrate how AoG mitigates cascading retrieval errors. EoG follows a locally plausible nationality-based path through the United States of America and eventually predicts the incorrect answer, Robert F. Kennedy. In contrast, AoG assigns this prefix a low recoverability score of 0.18, while assigning much higher scores to the family-related prefixes through Kennedy family and Rose Kennedy. This example demonstrates how prefix-state recoverability helps prevent early retrieval errors from propagating across subsequent hops. An additional case study on path explosion is provided in Appendix C.8.

Table 5: Case study of cascading error recovery.

Question	What relative of JFK was buried in St Peter's Church Edensor?
Gold Answer	**Kathleen Cavendish**
EoG	Reasoning Path: JFK → people.person.nationality → United States of America → people.person.nationality → Robert F. Kennedy **Answer:** Robert F. Kennedy
AoG	Reasoning Paths: (1) JFK → people.person.nationality → United States of America (2) JFK → people.family.members → Kennedy family → people.family_member.family → Kathleen Cavendish (3) JFK → people.person.parents → Rose Kennedy → people.person.children → Kathleen Cavendish
Recoverability Score	$v(s_{\text{United States of America}}) = 0.18$, $v(s_{\text{Kennedy family}}) = 0.92$, $v(s_{\text{Rose Kennedy}}) = 0.97$
AoG Prediction	**Answer:** Kathleen Cavendish **Reason:** The retained evidence connects JFK to Kathleen Cavendish via Kennedy family. The nationality-based prefix ending at United States of America receives a low recoverability score and is pruned. Therefore, AoG identifies Kathleen Cavendish as the correct relative of JFK.

5 Conclusion

In this paper, we propose AoG, a recoverability-aware framework for multi-hop knowledge graph retrieval in LLM reasoning. AoG tackles cascading error propagation and path explosion by jointly learning a local action policy and a prefix-state value function. The action policy guides question- and history-aware relation–entity selection, while the value function estimates whether an intermediate prefix remains recoverable within the remaining hop budget. During inference, uncertainty-aware beam search integrates action confidence and state recoverability to preserve plausible paths and prune redundant or unrecoverable branches. Experiments on public KGQA benchmarks show that AoG retrieves higher-quality paths and improves downstream LLM reasoning. The results highlight the importance of modeling intermediate retrieval states for reliable and efficient KG-enhanced reasoning. In future work, we will extend AoG to incomplete and continuously updated KGs while developing weakly supervised strategies for recoverability-aware retrieval.

AI USE STATEMENT

In this work, ChatGPT was used solely for language editing, text condensation, and improving the clarity and readability of the manuscript. We did not use generative AI tools to generate research ideas or experimental results, conduct experiments or data analysis, or replace scientific judgment and interpretation. All AI-assisted revisions were manually reviewed and verified by the authors. In particular, we checked that the revised text preserved the intended technical meaning, remained consistent with the underlying methods and experimental results, and did not introduce inaccurate or unsupported claims. We take responsibility for the final content of this work, including all text, claims, results, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work focuses on KG-enhanced large language model reasoning and is evaluated on publicly available datasets (WebQSP, CWQ, GrailQA, MetaQA) and models (Qwen, Llama, GPT-4o). The study does not involve human-subject experiments, sensitive user data, or personally identifiable information. All datasets and models used in this work are publicly available and are used in accordance with their respective licenses and terms of use. Although the proposed method aims to improve factual grounding by retrieving structured evidence from knowledge graphs, it may still produce incorrect or misleading answers when the underlying KG is incomplete, outdated, or contains biased information. Such limitations may propagate to downstream LLM reasoning. Therefore, the proposed method is intended for research purposes, and appropriate verification and human oversight should be exercised in applications where factual errors may have significant consequences.

REPRODUCIBILITY STATEMENT

We have taken multiple steps to ensure reproducibility. Details of the experimental setup, datasets, and evaluation metrics are provided in the main paper (Section 4.1) and Appendix B. Hyperparameter details of AoG are reported in Appendix C.2.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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*, 2024.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of Thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17682–17690, 2024.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 1247–1250, 2008.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Yuzheng Cai, Zhenyue Guo, Yiwen Pei, Wanrui Bian, and Weiguo Zheng. SimgRAG: Leveraging similar subgraphs for knowledge graphs driven retrieval-augmented generation. In *Findings of the Association for Computational Linguistics*, pp. 3139–3158, 2025.

Hanzhu Chen, Xu Shen, Qitan Lv, Jie Wang, Xiaoqi Ni, and Jieping Ye. SAC-KG: Exploiting large language models as skilled automatic constructors for domain knowledge graph. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 4345–4360, 2024.

Hanzhu Chen, Xu Shen, Jie Wang, Zehao Wang, Qitan Lv, Junjie He, Rong Wu, Feng Wu, and Jieping Ye. Knowledge graph finetuning enhances knowledge manipulation in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PalLM scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

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.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6491–6501, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yu Gu, Sue Kase, Michelle Vanni, Brian M. Sadler, Percy Liang, Xifeng Yan, and Yu Su. Beyond I.I.D.: Three levels of generalization for question answering on knowledge bases. In *Proceedings of the web conference 2021*, pp. 3477–3488, 2021.

Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. DeepRAG: Thinking to retrieve step by step for large language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 553–561, 2021.

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

Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. KG-GPT: A general framework for reasoning on knowledge graphs using large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9410–9421, 2023.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Mufei Li, Siqi Miao, and Pan Li. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *International Conference on Learning Representations*, 2025a.

Songze Li, Zhiqiang Liu, Zhengke Gui, Huajun Chen, and Wen Zhang. Enrich-on-Graph: Query-graph alignment for complex reasoning with LLM enriching. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7672–7692, 2025b.

Xiaoxi Li, Yujia Zhou, and Zhicheng Dou. Unigen: A unified generative framework for retrieval and question answering with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 8688–8696, 2024.

Chenxiao Lin, Ye Luo, KunHong Liu, and Qingqiang Wu. Stepwise contrastive reasoning for retrieval-augmented generation over knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15243–15251, 2026.

Haochen Liu, Song Wang, Yaochen Zhu, Yushun Dong, and Jundong Li. Knowledge graph-enhanced large language models via path selection. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6311–6321, 2024.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*, 2024.

Linhao Luo, Zicheng Zhao, Reza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. GFM-RAG: Graph foundation model for retrieval augmented generation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 36371–36405, 2025.

Jie Ma, Zhitao Gao, Qi Chai, Wangchun Sun, Pinghui Wang, Hongbin Pei, Jing Tao, Lingyun Song, Jun Liu, Chen Zhang, and Lizhen Cui. Debate on Graph: A flexible and reliable reasoning framework for large language models. In *Thirty-Ninth AAAI Conference on Artificial Intelligence*, pp. 24768–24776, 2025.

Costas Mavromatis and George Karypis. ReaRev: Adaptive reasoning for question answering over knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2447–2458, 2022.

Costas Mavromatis and George Karypis. GNN-RAG: graph neural retrieval for efficient large language model reasoning on knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 16682–16699, 2025.

Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1400–1409, 2016.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599, 2024.

Minbae Park, Hyemin Yang, Jeonghyun Kim, Kunsoo Park, and Hyunjoon Kim. ProgRAG: Hallucination-resistant progressive retrieval and reasoning over knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32674–32682, 2026.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. Sequence-to-sequence knowledge graph completion and question answering. In *Proceedings of the 60th annual meeting of the association for computational linguistics*, pp. 2814–2828, 2022.

Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. TransferNet: An effective and transparent framework for multi-hop question answering over relation graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4149–4158, 2021.

Yiheng Shu, Zhiwei Yu, Yuhan Li, Börje F. Karlsson, Tingting Ma, Yuzhong Qu, and Chin-Yew Lin. TIARA: Multi-grained retrieval for robust question answering over large knowledge base. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8108–8121, 2022.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4231–4242, 2018.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M Ni, Heung-Yeung Shum, and Jian Guo. Think-on-Graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*, 2024.

Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 641–651, 2018.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Kaehyun Um, KyuHwan Yeom, Haerim Yang, Minyoung Choi, Hyeongjun Yang, and Kyong-Ho Lee. RPO-RAG: Aligning small LLMs with relation-aware preference optimization for knowledge graph question answering. In *Proceedings of the ACM Web Conference 2026*, pp. 2513–2523, 2026.

Song Wang, Junhong Lin, Xiaojie Guo, Julian Shun, Jundong Li, and Yada Zhu. Reasoning of large language models over knowledge graphs with super-relations. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

Yike Wu, Yi Huang, Nan Hu, Yuncheng Hua, Guilin Qi, Jiaoyan Chen, and Jeff Z. Pan. CoTKR: Chain-of-thought enhanced knowledge rewriting for complex knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3501–3520, 2024.

Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Jun Zhao, and Kang Liu. Generate-on-Graph: Treat LLM as both agent and KG for incomplete knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18410–18430, 2024.

An Yang, Baosong Yang, Binyuan Hui, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024a.

Diji Yang, Jinmeng Rao, Kezhen Chen, Xiaoyuan Guo, Yawen Zhang, Jie Yang, and Yi Zhang. IM-RAG: Multi-round retrieval-augmented generation through learning inner monologues. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 730–740, 2024b.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822, 2023.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 201–206, 2016.

Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 5773–5784, 2022a.

Taolin Zhang, Chengyu Wang, Nan Hu, Minghui Qiu, Chengguang Tang, Xiaofeng He, and Jun Huang. DKPLM: Decomposable knowledge-enhanced pre-trained language model for natural language understanding. In *Thirty-Sixth AAAI Conference on Artificial Intelligence*, pp. 11703–11711, 2022b.

Yufeng Zhang, Xuepeng Wang, Lingxiang Wu, and Jinqiao Wang. Enhancing chain of thought prompting in large language models via reasoning patterns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25985–25993, 2025.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. Variational reasoning for question answering with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 6069–6076, 2018.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*, 2023.

A Detailed Retrieval Algorithm

Given a question q, topic entities $\mathcal{E}_q$, and KG $\mathcal{G}$, AoG performs recoverability-aware beam search to retrieve multi-hop reasoning paths. It maintains a beam of candidate reasoning prefixes and progressively expands them according to the learned action policy. Meanwhile, the value function estimates the recoverability of intermediate states, allowing the search to preserve promising prefixes that may still lead to useful evidence. The detailed pseudocode of the recoverability-aware path retrieval procedure is presented in Algorithm 1.

Algorithm 1 Recoverability-aware reasoning path retrieval

Require: Topic entities $\mathcal{E}_q$, KG $\mathcal{G}$, maximum hop H, beam width w, threshold δ, value weight α

Ensure: Reasoning paths Π

1: $B_0 \leftarrow \{[e] \mid e \in \mathcal{E}_q\}$, $\mathcal{P} \leftarrow \emptyset$
2: **for** $t = 0$ to $H - 1$ **do**
3: $\quad \mathcal{C} \leftarrow \emptyset$
4: $\quad$ **for all** $\pi_{<t} \in B_t$ **do**
5: $\quad\quad e_t \leftarrow$ last entity of $\pi_{<t}$, $s_t \leftarrow (q, e_t, \pi_{<t})$
6: $\quad\quad \mathcal{A}_t \leftarrow \{(r, e') \mid (e_t, r, e') \in \mathcal{G}\}$
7: $\quad\quad \mathcal{A}'_t \leftarrow \mathrm{UncertainExpand}(\mathcal{A}_t, s_t, p_\theta, \delta)$
8: $\quad\quad$ **for all** $a_t = (r_t, e_{t+1}) \in \mathcal{A}'_t$ **do**
9: $\quad\quad\quad \pi_{\le t} \leftarrow \pi_{<t} \oplus a_t$
10: $\quad\quad\quad s_{t+1} \leftarrow (q, e_{t+1}, \pi_{\le t})$
11: $\quad\quad\quad \mathrm{Score}(\pi_{\le t}) \leftarrow \frac{1}{t+1} \sum_{i=0}^{t} \log p_\theta(a_i \mid s_i) + \alpha \log v_\phi(s_{t+1})$
12: $\quad\quad\quad \mathcal{C} \leftarrow \mathcal{C} \cup \{\pi_{\le t}\}$, $\mathcal{P} \leftarrow \mathcal{P} \cup \{\pi_{\le t}\}$
13: $\quad\quad$ **end for**
14: $\quad$ **end for**
15: $\quad \mathcal{C} \leftarrow \mathrm{MergeSameEntity}(\mathcal{C})$
16: $\quad B_{t+1} \leftarrow \mathrm{TopW}(\mathcal{C}, w)$
17: **end for**
18: $\Pi \leftarrow \mathrm{TopK}(\mathcal{P})$
19: **return** Π

AoG initializes the beam with the topic entities and maintains a path pool for candidate evidence (line 1). At each hop, it enumerates outgoing relation–entity actions from the current entity and applies $\mathrm{UncertainExpand}(\cdot)$ to retain promising actions according to p_θ and the uncertainty threshold δ (lines 5–7). Each selected action extends the current prefix, and the resulting path is scored by combining its average policy confidence with the recoverability estimated by v_ϕ, weighted by α (lines 8–12). Prefixes ending at the same entity are then merged to reduce redundant exploration, and the top-w candidates are retained for the next hop (lines 15–16). Importantly, merging only affects the active beam, while all generated prefixes remain in the path pool, preserving diverse reasoning histories for subsequent evidence selection. Finally, the top-k paths accumulated in the path pool are returned as structured evidence for downstream LLM reasoning (lines 18–19).

B EXPERIMENTAL DETAILS

B.1 DATASETS

We conduct comparative experiments on two widely used benchmarks: WebQSP (Yih et al., 2016) and CWQ (Talmor & Berant, 2018). WebQSP contains 4,737 questions, mainly involving 1-hop and 2-hop reasoning, whereas CWQ includes approximately 34,689 more challenging questions that typically require up to 4-hop compositional reasoning. Both datasets use Freebase (Bollacker et al., 2008) as the underlying knowledge source, which contains around 88 million entities, 20 thousand relations, and 126 million triples. Table 6 summarizes the overall dataset statistics, while Table 7 presents the distribution of the numbers of answers.

Table 6: Statistics of datasets in WebQSP and CWQ.

Datasets	KB	#Train	#Test	1 hop	2 hop	$\geq$ 3 hop	Max #hop
WebQSP	Freebase	2826	1628	65.49%	34.51%	0.00%	2
CWQ	Freebase	27639	3531	40.91%	38.34%	20.75%	4

Table 7: Answer distribution proportion of WebQSP and CWQ.

Dataset	$\#\text{Ans} = 1$	$2 \leq \#\text{Ans} \leq 4$	$5 \leq \#\text{Ans} \leq 9$	$\#\text{Ans} \geq 10$
WebQSP	51.2%	27.4%	8.3%	12.1%
CWQ	70.6%	19.4%	6.0%	4.0%

B.2 EVALUATION METRIC DETAILS

For downstream KGQA performance, we use Hits@1 and F1 following prior work (Um et al., 2026; Li et al., 2025a). Hits@1 measures the proportion of questions for which the top-ranked prediction matches a gold answer. Since a question may have multiple valid answers, F1 additionally measures the overlap between predicted and gold answer sets by balancing precision and recall.

For retrieval evaluation, Hit@k measures whether the top-k retrieved results contain at least one gold answer entity. We further report Path Recall@k to evaluate path-level retrieval quality. Path Recall@k measures whether at least one of the top-k retrieved paths forms a complete answer-reaching path from a topic entity to any gold answer. Together, these metrics evaluate both answer coverage and the completeness of retrieved reasoning evidence.

B.3 BASELINES

We compare AoG with representative baselines grouped into four categories: 1) Embedding-based methods, 2) Retrieval-based methods, 3) LLMs, and 4) LLMs+KGs methods. The details of each baseline are described as follows.

Embedding-based methods.

- **KV-Mem** (Miller et al., 2016) adopts a key-value memory network to store knowledge graph triples and performs multi-hop reasoning through iterative memory operations.
- **NSM** (He et al., 2021) employs a sequential reasoning model to mimic the multi-hop reasoning process over knowledge graphs.
- **TransferNet** (Shi et al., 2021) uses a graph neural network to capture the relevance between entities and questions for reasoning.
- **KGT5** (Saxena et al., 2022) fine-tunes a sequence-to-sequence model over knowledge graphs and generates answers conditioned on the input question.

Retrieval-based methods.

- **GraftNet** (Sun et al., 2018) constructs a question-specific subgraph and applies a graph convolutional network to reason over KB entities and relations.
- **SR+NSM** (Zhang et al., 2022a) employs a trainable subgraph retriever to identify question-relevant paths and feeds the resulting subgraph into NSM for multi-hop reasoning.
- **SR+NSM+E2E** (Zhang et al., 2022a) further fine-tunes the subgraph retriever using feedback from the NSM reasoner, enabling retrieval and reasoning to mutually improve during training.
- **ReaRev** (Mavromatis & Karypis, 2022) adaptively updates reasoning instructions using KG-aware information and executes them with a GNN-based breadth-first search strategy.

Large language models.

- **Qwen2-7B** (Yang et al., 2024a) is a 7B-parameter open-weight model that serves as a mid-scale backbone for instruction-tuned reasoning.
- **Llama2-7B** (Touvron et al., 2023) serves as a representative open-source LLM baseline for evaluating general reasoning performance on KGQA tasks.
- **Llama3.1-8B** (Grattafiori et al., 2024) is included as a stronger open-source LLM baseline to examine the impact of improved backbone reasoning capability.
- **GPT-4o-mini** (Achiam et al., 2023) is a proprietary LLM developed by OpenAI and is included as a strong closed-source reasoning baseline.

LLMs+KGs methods.

- **ToG** (Sun et al., 2024) treats the LLM as an agent that iteratively explores knowledge graph relations and entities, using beam search to construct promising reasoning paths without additional model training.
- **RoG** (Luo et al., 2024) follows a planning–retrieval–reasoning pipeline that first generates relation-level plans, grounds them in the knowledge graph, and then uses the retrieved paths as evidence for answer generation.
- **GNN-RAG** (Mavromatis & Karypis, 2025) employs a lightweight GNN to identify answer-relevant knowledge graph nodes and extracts shortest paths connecting question entities to predicted answer candidates as context for the LLM.
- **ReKnoS** (Wang et al., 2025) introduces super-relations that aggregate multiple relational paths, enabling bidirectional knowledge graph exploration and improving the successful retrieval of reasoning evidence.
- **EoG** (Li et al., 2025b) leverages LLM prior knowledge to enrich the knowledge graph and improve alignment between the query and graph, thereby enabling more effective evidence extraction for downstream KGQA reasoning.
- **ProgRAG** (Park et al., 2026) decomposes complex questions into sub-questions and progressively extends partial reasoning paths with externally retrieved evidence, while applying uncertainty-aware pruning to retain reliable evidence for subsequent reasoning.
- **SCR** (Lin et al., 2026) performs stepwise contrastive alignment between the question and neighboring knowledge graph entities, iteratively selecting the most relevant entity and updating the question representation to navigate toward the answer.

C Additional Experiments

C.1 Efficiency Analysis

Table 8 compares the efficiency of AoG with representative baselines in terms of average retrieval time, generation time, total runtime, and the number of LLM calls. Compared with ToG, AoG substantially reduces the overall inference cost by avoiding repeated LLM interactions during iterative graph

exploration and invoking the LLM only once for final answer generation. Specifically, AoG reduces the total runtime from 18.7 s to 3.6 s on WebQSP and from 23.2 s to 5.5 s on CWQ. Compared with RoG, GNN-RAG, and EoG, AoG also achieves the lowest or comparable total runtime while requiring only a single LLM call. Although its retrieval time is slightly higher than that of GNN-RAG, AoG constructs more compact and focused evidence, resulting in substantially lower generation time. This efficiency is mainly attributed to two design choices: uncertainty-aware beam search selectively preserves plausible alternatives instead of exhaustively expanding the search space, while same-entity merging eliminates redundant prefixes that converge to the same intermediate entity. Consequently, fewer noisy or duplicated paths are passed to the final LLM, reducing both context size and generation overhead. Overall, AoG achieves efficient end-to-end reasoning by jointly reducing unnecessary graph exploration, redundant evidence, and repeated LLM interactions, while maintaining strong reasoning performance.

Table 8: Efficiency comparison of AoG and other baselines.

Method	WebQSP				CWQ			
	Retrieval Time (s)	Generation Time (s)	Total Time (s)	LLM Calls	Retrieval Time (s)	Generation Time (s)	Total Time (s)	LLM Calls
ToG	16.2	2.5	18.7	13.6	19.6	3.6	23.2	21.4
RoG	3.3	1.4	4.7	2	5.1	2.0	7.1	2
GNN-RAG	**2.8**	1.0	3.8	**1**	**4.4**	1.1	**5.5**	**1**
EoG	5.1	2.3	7.4	4	6.8	3.5	10.3	4
AoG	3.0	**0.6**	**3.6**	**1**	4.8	**0.7**	**5.5**	**1**

C.2 Parameter Sensitivity Analysis

We conduct a sensitivity analysis to evaluate the impact of key hyperparameters, including the number of retained candidate paths k, the beam size w, and the expansion threshold δ. The effects of k and w are shown in Figure 5. As k increases, the number of retrieved candidate paths and retrieval recall improve consistently, indicating that expanding the candidate path space helps cover more potential correct answers. However, excessively large k also introduces more noisy paths, which limits or even degrades the final reasoning performance. Similarly, increasing the beam size w enlarges the search space and improves recall, but it leads to higher computational overhead while providing only marginal performance gains. By balancing reasoning accuracy and efficiency, we ultimately set $k = 25$ and $w = 8$ on WebQSP, and $k = 25$ and $w = 12$ on CWQ.

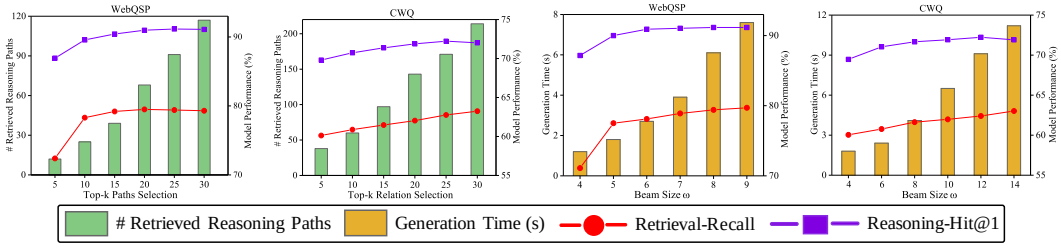


Figure 5: Sensitivity analysis of hyperparameters.

We further investigate the influence of the uncertainty threshold δ, which controls the adaptive expansion of candidate actions. As shown in Table 9, AoG achieves stable performance across different threshold values on both datasets. Specifically, the best results are obtained with $\delta = 0.1$, achieving 92.1 Hits@1 on WebQSP and 74.5 Hits@1 on CWQ. A smaller threshold ($\delta = 0.05$) restricts exploration and may discard useful alternative paths, while a larger threshold ($\delta \geq 0.15$) introduces more uncertain branches and increases search noise. These results demonstrate that moderate uncertainty-aware expansion provides an effective balance between exploration and retrieval precision.

Table 9: The impact of threshold δ on Hits@1 performance.

Method	WebQSP				CWQ			
	δ=0.05	δ=0.1	δ=0.15	δ=0.2	δ=0.05	δ=0.1	δ=0.15	δ=0.2
AoG	90.7	92.1	91.5	90.8	72.8	74.5	73.9	73.5

C.3 Effect of Retrieval Quality on WebQSP Performance

As shown in Table 10, AoG consistently achieves the strongest retrieval performance on WebQSP while using the most compact KG context. Compared with SCR, AoG improves Path Recall@10 from 78.0 to 82.6. In particular, the improvement in Path Recall@10 indicates that recoverability-aware retrieval is more effective at preserving complete answer-reaching reasoning paths rather than retaining only locally relevant evidence. Meanwhile, AoG reduces the average retrieved KG context from 240 to 186 tokens, a reduction of 22.5%, suggesting that its adaptive expansion and prefix pruning effectively suppress redundant exploration. The resulting evidence is therefore both more concise and more informative, which also translates into improved downstream reasoning performance, increasing Hits@1 from 90.8 to 92.1. These results demonstrate that AoG achieves a favorable balance between high-quality evidence and answer accuracy.

Table 10: Comparison of different retrieval methods on WebQSP.

Method	Retrieval				Reasoning
	#KG Tokens	Hit@1	Hit@10	Path Recall@10	Hits@1
ToG	3620	51.2	74.6	66.8	82.6
RoG	482	55.4	78.9	72.5	85.7
GNN-RAG	396	57.1	79.8	73.4	85.7
SCR	240	61.3	82.9	78.0	90.8
AoG	**186**	**63.8**	**85.1**	**82.6**	**92.1**

C.4 Performance on Additional Benchmark

We supplement our experiments with evaluations on the larger GrailQA (Gu et al., 2021) benchmark built on Freebase. Combined with the original experimental result in Table 1, AoG consistently achieves strong performance across KGQA benchmarks with different scales and reasoning complexities, demonstrating the robustness of the proposed retrieval framework. Specifically, as shown in Table 11, on GrailQA, AoG improves Hits@1 over TIARA (Shu et al., 2022), ToG, DoG (Ma et al., 2025), ReKnoS, and EoG by 14.5%, 7.8%, 7.5%, 7.0%, and 2.0%, respectively, while also outperforming TIARA and EoG by 3.8% and 2.7% in F1. These results further demonstrate the consistent effectiveness of AoG across KGQA benchmarks with different scales and reasoning complexities.

Table 11: Performance comparison of advanced methods on GrailQA. The best results are in **bold**.

Method	GrailQA	
	Hits@1	F1
TIARA	73.0	78.5
ToG	79.7	–
DoG	80.0	–
ReKnoS	80.5	–
EoG	85.5	79.6
AoG	**87.5**	**82.3**

C.5 Transferability to Other KGs

We evaluate the transferability of AoG across different KGs using MetaQA-2hop and MetaQA-3hop (Zhang et al., 2018), which are built on the Wiki-Movies KG. We randomly sample 1,000 training

examples and consider two training strategies to fine-tune AoG: (1) *training from scratch*, where AoG is trained directly with Llama3.1-8B on the sampled MetaQA data, and (2) *transfer from Freebase*, where the AoG model previously trained on Freebase is further fine-tuned on the same MetaQA samples. As shown in Table 12, transfer from Freebase consistently outperforms training from scratch, improving Hits@1 by 6.5% and 3.4% on MetaQA-2hop and MetaQA-3hop, respectively, and F1 by 2.6% and 7.5%. These results indicate that the retrieval knowledge learned by AoG can be effectively transferred across heterogeneous knowledge graphs.

Table 12: Performance comparison of advanced methods on MetaQA-2hop and MetaQA-3hop. The best results are highlighted in **bold**, and results marked with * are obtained through reproduction.

Method	MetaQA-2hop		MetaQA-3hop	
	Hits@1	F1	Hits@1	F1
LLM Methods				
Llama2-7B	43.4	21.7	22.8	16.4
Llama3.1-8B	50.1	38.5	30.4	16.6
GPT-4o-mini	56.6	42.1	38.8	17.5
LLM+KG Methods				
KG-GPT* (Kim et al., 2023)	94.1	52.5	73.1	25.3
KELP* (Liu et al., 2024)	85.9	55.7	66.7	25.0
DoG	91.0	–	84.8	–
AoG (train from scratch)	88.3	83.0	85.7	44.2
AoG (transfer from Freebase)	**94.8**	**85.6**	**89.1**	**51.7**

We also compare the training cost of AoG on Freebase with the cost of transferring it to Wiki-Movies. As shown in Table 13, training AoG on Freebase takes 25 hours, while transferring it to Wiki-Movies requires only 1.6 hours, corresponding to a 93.6% reduction in training time. These results demonstrate that AoG can be efficiently adapted to a new KG without the overhead of full retraining.

Table 13: Training time comparison.

Method	Training on Freebase	Transferring to Wiki-Movies
AoG	25 hours	1.6 hours

C.6 Performance On Different Hops

We analyze the performance of AoG across questions with different reasoning hops in Table 14. While AoG performs comparably to the strongest baselines on 1-hop questions, its advantage becomes more pronounced as the reasoning hop increases. On WebQSP, AoG achieves an F1 score of 72.2 on 2-hop questions, outperforming GNN-RAG and SCR by 2.4% and 0.7%, respectively. A similar trend is observed on CWQ, where AoG achieves the best performance on both 2-hop and $\geq$3-hop questions, with F1 scores of 72.5 and 65.3. These results indicate that the benefit of recoverability-aware retrieval becomes more evident for deeper multi-hop reasoning, where early retrieval errors are more likely to propagate across subsequent hops.

C.7 Performance On Different Answer Numbers

We further analyze the performance of different methods with respect to the number of answers in Table 15. AoG achieves the highest F1 scores in most answer-count groups, demonstrating robust performance across varying answer-set sizes. On WebQSP, AoG performs particularly well on

Table 14: Performance analysis (F1) based on the number of maximum hops.

Method	WebQSP			CWQ		
	1 hop	2 hop	$\geq$ 3 hop	1 hop	2 hop	$\geq$ 3 hop
RoG	77.0	64.9	–	62.9	58.5	37.8
GNN-RAG	72.0	69.8	–	47.4	69.4	51.8
SCR	77.2	71.5	–	50.4	70.1	64.2
AoG	79.8	72.2	–	58.8	72.5	65.3

questions with fewer than 10 answers and remains competitive even when the answer set contains 10 or more entities. On CWQ, AoG consistently surpasses RoG and GNN-RAG across all answer-count groups. While questions with larger answer sets are generally more challenging, the performance degradation of AoG is relatively limited, allowing it to retain a clear advantage over competing methods. These results indicate that recoverability-aware retrieval enables AoG to preserve more informative and reliable reasoning paths, thereby providing stronger evidence for multi-answer reasoning.

Table 15: Performance analysis (F1) based on the number of answers (#Ans).

Method	WebQSP				CWQ			
	#Ans = 1	$2 \leq$ #Ans ≤ 4	$5 \leq$ #Ans ≤ 9	#Ans ≥ 10	#Ans = 1	$2 \leq$ #Ans ≤ 4	$5 \leq$ #Ans ≤ 9	#Ans ≥ 10
RoG	67.9	79.4	75.0	58.3	56.9	53.7	58.4	43.6
GNN-RAG	71.2	76.3	74.1	56.3	60.4	55.5	61.5	50.1
AoG	74.5	82.8	77.3	58.7	64.9	57.0	63.2	50.6

C.8 Case Study

Table 16 illustrates the reduction of path explosion. EoG retrieves several redundant branches involving books and influenced people, enlarging the search space and introducing noisy evidence. AoG instead retains compact and high-recoverability evidence paths, thereby reducing redundant exploration while still providing sufficient support for the correct prediction.

Table 16: Case study of path explosion reduction.

Question	In which city within Kingsbury County did Laura Ingalls Wilder live?
Gold Answer	**De Smet**
EoG	Reasoning Paths: (1) Laura Ingalls Wilder → influence.influence_node.influenced → Rose Wilder Lane → book.author.works_written → The Making Of Herbert Hoover (2) Laura Ingalls Wilder → book.book_edition.author_editor → The Long Winter → fictional_universe.fictional_setting.works_set_here → De Smet . . . . . . (10) Laura Ingalls Wilder → people.person.parents → Caroline Ingalls → people.person.place_of_birth → Brookfield **Answer:** De Smet
AoG	Reasoning Paths: (1) Laura Ingalls Wilder → people.place_lived.person → place-lived record → people.place_lived.location → De Smet (2) Laura Ingalls Wilder → people.person.parents → Caroline Ingalls → people.deceased_person.place_of_death → De Smet (3) Kingsbury County → location.location.containedby → De Smet
Expanded Paths	**EoG: 10 candidate paths** **AoG: 3 candidate paths**
AoG Prediction	**Answer:** De Smet **Reason:** The retained evidence identifies De Smet as the city associated with Laura Ingalls Wilder. Compared with redundant branches, AoG provides compact evidence for prediction.