TOWARDS A TRANSFERABLE FOUNDATION RETRIEVER FOR MULTI-HOP REASONING OVER KNOWLEDGE GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Foundation retrievers aim to reuse retrieval knowledge across knowledge graphs (KGs), reducing the need to train a separate model for each graph. For multi-hop reasoning, this requires identifying relevant relations and ranking answers under changes in graph structure and question distribution. Yet strong in-domain performance does not necessarily translate into effective cross-KG retrieval. Our analysis of KGFR reveals a substantial ranking gap: correct answers often remain reachable but receive low ranks after transfer. We propose KGSEEK, a multi-source foundation retriever that uses semantic relevance to guide retrieval and graph structure to distinguish candidate answers. KGSEEK learns question–relation relevance from source questions and answers. It aggregates this relevance along paths to form a semantic ranking reference. A shared scorer incorporates neighborhood context through bounded adjustments. Length-normalized path scoring preserves this bound at the entity level, allowing contextual refinement while retaining strong semantic preferences. Joint training across source KGs learns shared scoring functions that apply to new graphs through their relation labels and structure. Experiments on Freebase and DBpedia show that one source-trained model improves cross-KG retrieval and downstream question answering (QA) over strong baselines with different large language model (LLM) readers. The same architecture also improves in-domain retrieval and maintains competitive QA performance. These results show the value of semantic guidance for transferable multi-hop retrieval.

1 INTRODUCTION

Multi-hop reasoning over knowledge graphs (KGs) requires connecting facts across relations to identify relevant answers. A retriever supports this process by finding relational evidence and ranking candidate entities. Many knowledge graph question answering (KGQA) retrievers learn from questions and answers associated with a particular graph. Deploying them on another KG often requires additional supervision. A *foundation retriever* aims to reduce this dependence: it should learn from available KGs and apply its retrieval knowledge to new graphs. Recent KG foundation models, including ULTRA and KG-ICL, have advanced reasoning over unseen entities and relations (Galkin et al., 2024; Cui et al., 2024). KGFR extends this direction to retrieval for natural-language questions (Cui et al., 2026).

A key question remains: how well does a foundation retriever preserve its ranking ability across KGs? Strong in-domain performance alone does not establish this ability. Different graphs introduce new relation vocabularies, neighborhood structures, and question distributions. We examine this challenge through a cross-dataset analysis of KGFR (Cui et al., 2026) with matched training sizes and hop distributions. As Figure 1 shows, a model trained on KQA Pro retains only 9.9% and 18.4% of the corresponding in-domain Hits@1 on WebQSP and ComplexWebQuestions (CWQ), respectively. Correct answers nevertheless remain reachable for most questions. The main gap here lies in ranking: the retriever can reach a correct answer but places incorrect candidates ahead of it.

This observation motivates a simple design principle: *semantic relevance should guide retrieval, while graph structure should help distinguish candidate answers*. Relation labels describe the meaning of graph edges and provide a natural basis for transfer. For example, questions about a person's

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.

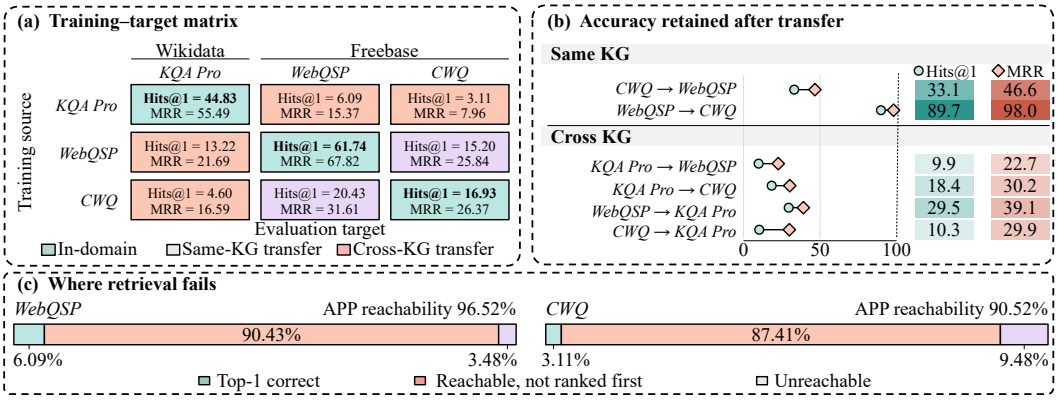

Figure 1: KGFR's cross-KG ranking gap. Validation Hits@1 / MRR (%): (a) training–target pairs; (b) scores retained relative to each target's in-domain results. (c) KQA Pro-to-Freebase outcomes separate ranking errors from unreachable answers. Appendix A details this analysis.

education call for education-related evidence across different KGs. Source questions and answers can teach a model these semantic correspondences. Neighborhood structure supplies complementary information when candidates have similar relation paths. However, neighborhood patterns can change across graphs. The model should therefore use context to refine semantic preferences while preserving sufficiently strong ones.

We introduce KGSEEK, a multi-source foundation retriever built around this principle. KGSEEK learns question–relation relevance from source answers and combines relation scores along paths to form a semantic ranking reference. A shared scorer refines this reference through bounded contextual adjustments. The model averages edge scores along each path and ranks entities by their highest-scoring path. This aggregation preserves the adjustment bound at the entity level, independently of path length. Context can thus resolve close semantic comparisons without overturning sufficiently strong reference preferences. Source QA supervision trains the reference and contextual rankings jointly. The resulting model applies to new graphs through their relation labels and structure, without additional target-QA training.

We train one model on KQA Pro and MetaQA and evaluate its transfer to Freebase and DBpedia. KGSEEK improves retrieval over KGFR (Cui et al., 2026), GFM-RAG (Luo et al., 2025b), and G-reasoner (Luo et al., 2026) on WebQSP, CWQ, and LC-QuAD 1.0. These gains also improve downstream QA with two LLM readers. With in-domain training, the same architecture achieves stronger retrieval and competitive QA performance.

Our contributions are threefold:

- We identify answer ranking as a key bottleneck in cross-KG retrieval. Our analysis of KGFR shows that correct answers often remain reachable after transfer but rank poorly. This reveals a substantial gap between answer reachability and ranking accuracy.
- We propose KGSeek, which learns a semantic ranking reference from source answers and refines it through bounded contextual scoring. We show that maximum-mean path aggregation preserves the adjustment bound at the entity level, independently of path length.
- Experiments on three datasets across two unseen KGs show that KGSEEK outperforms strong retrieval baselines and improves QA with two LLM readers. With in-domain training, the same architecture achieves stronger retrieval and competitive QA performance.

2 RELATED WORK

Transferable knowledge graph reasoning. Multi-source KG pretraining transfers knowledge to target graphs through local retraining and distillation (Sun et al., 2023). Transfer-and-Fusion

integrates knowledge across KG schemas for cross-KG link prediction (Cui et al., 2025). Inductive KG reasoning generalizes to unseen entities through local subgraphs (Teru et al., 2020) and to unseen relations through relation representations (Lee et al., 2023; Galkin et al., 2024). Prompt graphs and random walks also enable transfer (Cui et al., 2024; Kim et al., 2026). MOTIF studies model expressivity (Huang et al., 2025), while SEMMA combines relation semantics with graph structure for inductive link prediction (Arun et al., 2025). We extend this work to natural-language retrieval, learning to preserve answer rankings across changes in both graph structure and question distribution.

Retrieval for multi-hop question answering. Multi-hop KGQA combines evidence retrieval with graph reasoning. Early methods construct or expand question-specific subgraphs (Sun et al., 2018; 2019). Later methods separate retrieval from reasoning (Zhang et al., 2022), share question–relation matching and propagation (Jiang et al., 2023), or combine graph transformation with path-based reasoning for efficient cross-dataset retrieval (Yao et al., 2025). KGFR, our closest baseline, couples language-derived relation representations with question-aware graph reasoning and asymmetric progressive propagation (APP) (Cui et al., 2026). We retain APP but learn a distinct ranking function: source answers supervise semantic relevance, and bounded graph context refines candidate rankings.

Graph-grounded language model reasoning. Graph-grounded QA uses LLMs to explore, plan, or constrain KG paths (Sun et al., 2024; Luo et al., 2024; 2025a), or learns to retrieve graph evidence (Mavromatis & Karypis, 2025; Li et al., 2025; He et al., 2024). ChainRAG completes missing entities in sub-questions and retrieves sentence-graph evidence for multi-hop QA (Zhu et al., 2025). GraphFlow learns retrieval policies from outcome rewards (Yu et al., 2025). GFM-RAG learns transferable retrieval over document-derived graphs (Luo et al., 2025b). G-reasoner combines graph and language foundation models for heterogeneous knowledge. It trains graph predictions with answer supervision and distillation from frozen text-encoder relevance (Luo et al., 2026). KGSEEK learns an explicit semantic ranking reference from source answers. This learned reference guides bounded contextual refinement through a shared scoring head.

Graph encoders and path-based scoring. Graph Transformers encode structural context (Ying et al., 2021; Rampášek et al., 2022; Liu et al., 2024), while differentiable graph propagation supports link prediction and multi-hop QA (Zhu et al., 2021; Shi et al., 2021). KGSEEK combines a sparse Graph Transformer with maximum-mean path scoring. Path averaging preserves the contextual adjustment bound at the entity level, independently of path length.

3 A MULTI-SOURCE FOUNDATION RETRIEVER

As shown in Figure 2, our KGSEEK uses semantic relevance to guide retrieval and graph context to distinguish candidate answers. A semantic reference learns question–relation relevance from source QA labels. A shared scorer then uses neighborhood features to refine these edge weights within a bounded range. We average edge weights along each path and rank entities by their best path. This aggregation preserves the context bound at the entity level. Joint supervision trains the semantic reference and final ranking across source KGs.

3.1 PROBLEM FORMULATION

Retrieval for multi-hop reasoning. We study entity retrieval for multi-hop reasoning in KGQA. Given a question and its topic entities, the retriever ranks candidate answers through relational paths in a KG. Let $G = (V, R, E)$ denote a directed, relation-labeled KG. We add the inverse (v, r^{-1}, u) of each fact (u, r, v) to support traversal in both directions. The sets R and E include these inverse relations and edges. A query consists of a natural-language question q and supplied topic entities $T \subseteq V$. The annotated gold set Y may include entities outside V. The retriever uses graph structure and readable relation labels to score candidates in a question-specific support graph. Paths connect topic entities to candidates through one or more edges.

Multi-source learning and transfer. We train one shared model on M source graphs $\{G_s\}_{s=1}^{M}$ and their QA training sets $\{\mathcal{D}_s\}_{s=1}^{M}$:

$$\min_{\theta} \frac{1}{M} \sum_{s=1}^{M} \mathbb{E}_{(q,T,Y)\sim\mathcal{D}_s} \mathcal{L}(\theta; q, T, Y, G_s). \tag{1}$$

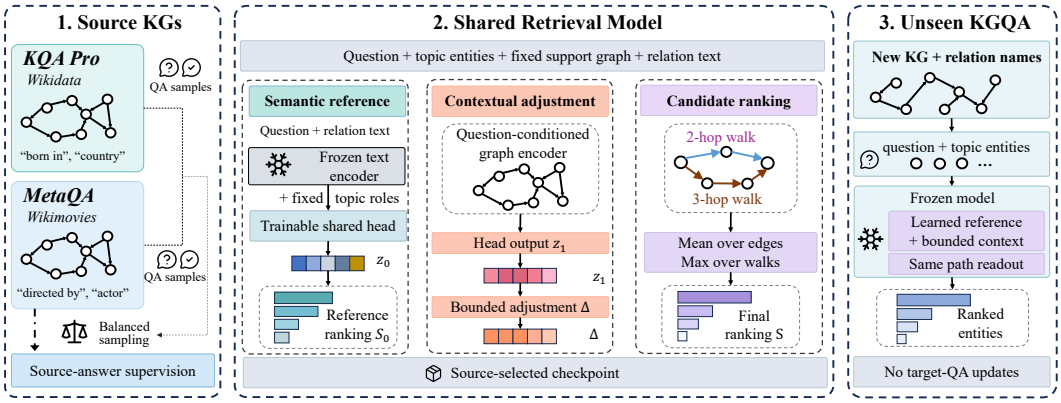

Figure 2: Overview of KGSEEK. A shared head produces z_0 from topic-role inputs and z_1 from neighborhood inputs. Their difference yields a bounded adjustment to z_0 before edge scoring and maximum-mean walk aggregation. Source answers jointly supervise S_0 and S_θ. The same trained parameters transfer through a new KG's relation labels and structure. In the figure, $S = S_\theta$.

Here θ contains all trainable parameters. The objective assigns equal weight to each source. At deployment, the model applies the same parameters to a held-out target graph $G_\star$. It interprets the target relations through their text labels and reasons over the target structure. Source QA data provide the training supervision. The target KG provides the graph and relation vocabulary for inference.

3.2 Shared Graph and Text Representations

For question q with topic entities T, KGFR's asymmetric progressive propagation (APP) supplies a bounded-hop support graph $\mathcal{G}_q = (\mathcal{V}_q, \mathcal{E}_q)$ (Cui et al., 2026). We take the union of directed KG edges that APP retains over H steps. All L encoder layers reuse this support. The retrieval horizon H and encoder depth L serve separate roles. We add artificial self loops for graph encoding and use the KG edges for path scoring.

A frozen text encoder represents questions and relation labels in a shared embedding space. Each entity starts with a topic-role feature: $\mathbf{t}_v = \mathbf{1}_d$ for $v \in T$ and $\mathbf{0}_d$ otherwise, where d is the state dimension. These features let the model identify topic entities across KGs. A sparse, question-conditioned Graph Transformer encodes their neighborhoods into node features $\mathbf{h}_v^{(L)}$. The question conditions attention queries. Relation features contribute to attention keys and values. All source KGs share the encoder parameters. Appendix B.1 gives the attention equations.

3.3 Semantic Scoring with Bounded Graph Context

The semantic reference scores each edge from the question, relation text, and endpoint topic roles. Source answers supervise these scores through path-based entity ranking. Neighborhood features provide complementary information. For example, two edges may connect a topic entity to different non-topic candidates through the same relation. The reference assigns these edges equal weights. Different neighborhoods can then help the shared scorer distinguish the candidates.

Semantic reference. Let $\bar{\mathbf{q}}$ and $\bar{\mathbf{r}}$ denote unit-normalized frozen embeddings of the question and relation text. The relation text specifies the traversal direction. The scoring head uses these fixed embeddings. The graph encoder maintains separate layer-wise relation states (Appendix B.1). Following cosine-based sentence representations (Reimers & Gurevych, 2019), we define the fixed semantic score $c(q, r) = \bar{\mathbf{q}}^\top \bar{\mathbf{r}}$. For edge (u, r, v), the head $f_\psi(\mathbf{x}_u, \mathbf{x}_v, q, r)$ combines endpoint features with the question and relation embeddings. Its parameters ψ form part of the full parameter set θ. The semantic reference uses the topic roles as endpoint features:

$$z_0(q, u, r, v) = f_\psi(\mathbf{t}_u, \mathbf{t}_v, q, r), \qquad w_0(q, u, r, v) = c(q, r) + B \tanh z_0(q, u, r, v). \tag{2}$$

$B > 0$ bounds the learned change from cosine similarity. The scorer learns this change from source answers while the text embeddings remain fixed. Edges with the same relation and topic-role inputs share a reference weight across entities. We initialize the head's output layer to zero, so the initial edge weights equal cosine similarity. Appendix B.2 specifies the head architecture.

Bounded contextual adjustment. The same head scores edges with contextual endpoint features: $z_1(q, u, r, v) = f_\psi(\mathbf{h}_u^{(L)}, \mathbf{h}_v^{(L)}, q, r)$. Sharing the head isolates the change in its inputs. The difference $z_1 - z_0$ measures how neighborhood features change its output relative to the topic roles. We convert this difference into a bounded adjustment Δ, with $\kappa > 0$ controlling its range:

$$\Delta(q, u, r, v) = \kappa \tanh\left(\frac{z_1(q, u, r, v) - z_0(q, u, r, v)}{\kappa}\right), \qquad (3)$$
$$w_\theta(q, u, r, v) = c(q, r) + B \tanh(z_0(q, u, r, v) + \Delta(q, u, r, v)).$$

The two bounds control different changes. B limits the final edge weight's deviation from cosine similarity. κ limits the contextual adjustment to the head output. Together, they bound the change from w_0 to w_θ by $B\kappa$. At $\kappa = 0$, we define $\Delta = 0$ and recover w_0. As κ grows without bound, Δ approaches $z_1 - z_0$ and the score approaches $c + B \tanh z_1$. The bound B still limits its deviation from cosine similarity.

3.4 Path-Based Entity Ranking

Path aggregation converts edge relevance into answer rankings. Summing edge scores would allow contextual adjustments to accumulate with path length. We instead average edge scores and select the highest-scoring path for each candidate. Let $\mathcal{P}_H(T, e)$ contain the directed walks from T to entity e in $\mathcal{G}_q$, with lengths $1, \ldots, H$. These walks follow fact edges and their inverses. They may revisit vertices or edges; artificial encoder self loops remain outside this set. For edge weights w, we define

$$\mathcal{R}(w; q, e) = \max_{p \in \mathcal{P}_H(T,e)} \frac{1}{|p|} \sum_{a \in p} w(q, a), \qquad S_\theta = \mathcal{R}(w_\theta), \quad S_0 = \mathcal{R}(w_0), \quad A = \mathcal{R}(c). \qquad (4)$$

Here a denotes a directed edge, and c assigns its relation's cosine score. The three scores use the same walks: S_θ gives the final ranking, S_0 gives the learned semantic reference, and A uses frozen cosine similarity. We assign $-\infty$ to unreachable entities. A single set of edge weights serves all walk lengths. Dynamic programming computes the scores in $O(H(|\mathcal{V}_q| + |\mathcal{E}_q|))$ (Appendix B.3). Inference ranks candidates by S_θ.

On a fixed support graph, averaging edge scores gives a path-length-independent bound on how context changes entity scores.

Proposition 1 (Score bounds on fixed support)**.** *For every reachable entity, the scoring rule satisfies $|S_\theta - A| \le B$ and $|S_0 - A| \le B$. It also satisfies $|S_\theta - S_0| \le B\kappa$. Thus the change in any pairwise gap relative to S_0 is at most $2B\kappa$, independent of path length. A reference winner with a gap greater than $2B\kappa$ over every other candidate remains the winner.*

Maximum-mean aggregation preserves the edge-level change bound: $\|\mathcal{R}(w) - \mathcal{R}(w')\|_\infty \le \|w - w'\|_\infty$ over reachable candidates. The 1-Lipschitz property of $\tanh$ then gives the stated bounds (Appendix B.4). On the same support, reference preferences with sufficient gaps persist. Path averaging makes the bound independent of path length.

3.5 Ranking Objectives and Multi-Source Training

Source answers directly supervise both the final ranking S_θ and the semantic reference S_0. We use a finite-margin objective to rank a correct answer above hard negatives. Once a comparison meets the margin, its loss vanishes. Training then focuses on the remaining comparisons.

Let $\mathcal{C}_q$ contain the reachable entities eligible as answers under the dataset mask (Appendix C), and let $Y_q = Y \cap \mathcal{C}_q$. For a query with $Y_q \neq \varnothing$ and either learned score $F \in \{S_\theta, S_0\}$, define

$$y_F^* = \arg\max_{y \in Y_q} F(q, y), \qquad \mathcal{N}_F = \mathrm{TopK}_{e \in \mathcal{C}_q \setminus Y_q} F(q, e) \cup \mathrm{TopK}_{e \in \mathcal{C}_q \setminus Y_q} A(q, e),$$
$$\mathcal{L}_{\mathrm{rank}}(F) = \frac{1}{|\mathcal{N}_F|} \sum_{n \in \mathcal{N}_F} \left[\max\left(0, \frac{m - [F(q, y_F^*) - F(q, n)]}{\tau}\right)\right]^2. \qquad (5)$$

Here $m > 0$ is the required margin, and $\tau > 0$ scales the loss. TopK selects up to K highest-scoring negative entities under each score. The union combines hard negatives from the learned ranking with semantically plausible alternatives from A. Each ranking selects its own highest-scoring gold answer y_F^*. The objective therefore favors placing at least one correct answer first, consistent with Hits@1 and MRR. We set the ranking loss to zero when the negative set is empty.

The path-ranking loss updates the graph encoder through the highest-scoring walks. An auxiliary listwise entity loss $\mathcal{L}_{\text{aux}}$ supplies direct answer supervision over eligible encoder nodes (Appendix B.5). For each query with a reachable eligible answer, the joint objective is

$$\mathcal{L} = \mathcal{L}_{\text{rank}}(S_\theta) + \beta \mathcal{L}_{\text{rank}}(S_0) + \eta \mathcal{L}_{\text{aux}}. \tag{6}$$

β weights reference supervision, and η weights the auxiliary loss. The auxiliary head supports training; the path score S_θ determines the deployed ranking.

We optimize this objective across source KGs according to Eq. (1). Balanced sampling alternates between equally sized source datasets. Source questions, topic entities, and answer labels train the shared parameters. The resulting model uses the same graph encoding and scoring functions on a new KG. Appendix D gives the sampling and answer-eligibility details.

4 EXPERIMENTS

We address three research questions. **RQ1:** How well does KGSEEK transfer retrieval and QA to unseen KGs? **RQ2:** How do reference supervision and bounded context contribute to transfer? **RQ3:** How well does the same architecture perform with in-domain supervision?

4.1 EXPERIMENTAL SETUP

Datasets. We use KQA Pro (Cao et al., 2022) and two-hop MetaQA (WikiMovies) (Zhang et al., 2018) as source datasets. KQA Pro provides a KG derived from Wikidata. We select KQA Pro entity-path questions and sample MetaQA to obtain 1,364 training and 174 validation examples per source. The target datasets are WebQSP and CWQ (Freebase) (Yih et al., 2016; Talmor & Berant, 2018) and LC-QuAD 1.0 (DBpedia) (Trivedi et al., 2017). WebQSP and CWQ use the full RoG-derived splits (Luo et al., 2024). LC-QuAD uses nonempty entity-answer queries from the official splits and a fixed DBpedia 2016-04 snapshot. Table 1 lists the dataset statistics. Appendix D details their construction.

Table 1: Dataset statistics. Target training and validation splits serve in-domain experiments.

Role	Dataset	KG family	Train	Validation	Test
Source	KQA Pro	Wikidata	1,364	174	—
	MetaQA	WikiMovies	1,364	174	—
Target	WebQSP	Freebase	2,826	246	1,628
	CWQ	Freebase	27,639	3,519	3,531
	LC-QuAD	DBpedia	2,662	296	740

Evaluation settings. In *cross-KG evaluation*, we train one retriever jointly on the two source datasets. We keep its parameters fixed and evaluate it on all three target test sets from unseen KGs. In *in-domain evaluation*, we train a separate retriever on each target dataset's training split and evaluate it on the same dataset's test split.

Evaluation protocol. We give all methods the same questions, oracle topic entities, background graphs, APP candidate subgraphs, and answer-eligibility rules. For cross-KG Freebase evaluation, we use the union of WebQSP and CWQ graph facts. For in-domain evaluation, we use each dataset's background graph. LC-QuAD uses the fixed DBpedia snapshot (Appendix D.1). We use readable relation labels and frozen BGE-large-en-v1.5 embeddings (Beijing Academy of Artificial Intelligence, 2023) throughout. APP uses $\lambda = 10$ and $H = 3$.

Baselines. We compare with three graph-based retrievers: KGFR (Cui et al., 2026), GFM-RAG (Luo et al., 2025b), and G-reasoner (Luo et al., 2026). KGFR learns question-conditioned graph reasoning with a listwise objective. GFM-RAG combines KG-completion pretraining with QA supervision. G-reasoner integrates graph reasoning, entity semantics, and semantic distillation. We adapt GFM-RAG and G-reasoner to entity retrieval on the supplied KG, denoted by †. All methods use the same QA training examples within each setting; GFM-RAG additionally uses KG triples for pretraining. Appendix C gives the adaptation details.

Table 2: Retrieval (%). Bold values indicate the best score for each metric, dataset, and setting.

Setting	Method	Freebase						DBpedia		
		WebQSP			CWQ			LC-QuAD		
		H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
Cross-KG	KGFR	9.28	30.90	16.55	2.95	14.05	6.71	17.97	51.62	29.22
	GFM-RAG†	6.94	28.56	13.85	1.64	8.33	3.87	12.30	46.49	22.98
	G-reasoner†	8.72	22.24	13.49	2.18	9.26	5.01	7.43	25.95	13.68
	KGSEEK	**37.10**	**69.90**	**47.63**	**11.92**	**52.05**	**25.24**	**32.43**	**75.14**	**45.96**
In-domain	KGFR	66.77	85.44	73.17	53.02	75.11	60.87	66.08	84.86	72.82
	GFM-RAG†	70.52	85.87	76.08	49.76	64.91	55.42	70.14	86.22	76.06
	G-reasoner†	67.75	84.15	74.01	50.07	70.89	57.40	71.62	87.30	77.47
	KGSEEK	**74.20**	**87.90**	**79.47**	**53.13**	**76.86**	**62.07**	**74.32**	**89.19**	**79.89**

Table 3: Downstream QA F1 with two readers (%). DeepSeek denotes DeepSeek V4 Flash. Qwen denotes Qwen3.6-27B. Bold marks the best score per dataset, reader, and setting.

Setting	Retriever	WebQSP		CWQ		LC-QuAD	
		DeepSeek	Qwen	DeepSeek	Qwen	DeepSeek	Qwen
Control	No retrieval	40.60	33.74	31.35	22.96	32.21	22.36
Cross-KG	KGFR	52.79	48.67	37.80	34.15	55.35	51.62
	GFM-RAG†	51.42	47.41	34.58	30.90	54.20	50.81
	G-reasoner†	47.87	43.52	33.32	30.61	51.51	46.94
	KGSEEK	**62.73**	**58.65**	**49.49**	**45.53**	**61.98**	**56.55**
In-domain	KGFR	66.72	60.87	54.25	49.46	64.22	57.93
	GFM-RAG†	**68.47**	**62.16**	51.56	47.28	64.63	58.22
	G-reasoner†	**68.47**	62.06	52.38	47.66	65.84	59.32
	KGSEEK	67.59	61.98	**55.69**	**51.26**	**66.74**	**61.77**

Implementation details. KGSEEK uses a three-layer graph encoder with 1,024-dimensional hidden states. We train it with Adam at a learning rate of 5×10^{-5}. We select the cross-KG checkpoint by average Hits@1 on the source validation sets and apply it to all three target datasets. For in-domain evaluation, we select checkpoints on each target's validation set. Appendix C specifies the hyperparameters and model-selection protocol.

Metrics and readers. We measure retrieval with Hits@1 (H@1), Hits@10 (H@10), and mean reciprocal rank (MRR), and QA with macro answer-set F1. We use DeepSeek V4 Flash and Qwen3.6-27B as readers. For each retriever, we give both readers the same prompt and evidence: up to 20 candidates and their KG paths. We construct these paths with the same rule for all retrievers. The no-retrieval baseline receives the question and topic entities. QA ablations use DeepSeek V4 Flash. We report percentages and express gains in percentage points (pp). Appendices E.1 and F provide additional metrics and reader configurations.

4.2 CROSS-KG GENERALIZATION

Retrieval performance. KGSEEK achieves the highest cross-KG Hits@1, Hits@10, and MRR on all three datasets (Table 2). The same source-trained model outperforms KGFR (Cui et al., 2026), the strongest baseline, on both Freebase and DBpedia. The gains therefore extend across target relation schemas. All methods have identical answer reachability on each dataset. The improvement comes from ranking answers in the shared candidate subgraphs, addressing the ranking gap in Figure 1.

Downstream QA. KGSEEK achieves the best cross-KG QA F1 on all three datasets with both readers (Table 3). We keep the reader and prompt fixed within each comparison, so the gains reflect changes in retrieved evidence. Both readers benefit across the two target KGs. This consistency shows that the retrieval advantage translates into better answers with different readers. Appendix E.2 reports paired confidence intervals for these gains.

Answer coverage. Broader answer coverage helps explain the QA gains beyond top-1 accuracy. On 1,263 CWQ questions, both methods rank an incorrect entity first, but only KGSEEK retrieves a gold answer in the top 20. This group's paired F1 differences account for most of the DeepSeek F1 gain (9.77 of 11.69 points). These results link broader answer coverage with better QA, even when top-1 predictions remain incorrect. Answer Recall@20 also rises on all three targets (Appendix E.3).

4.3 ABLATION STUDY

We isolate reference supervision and the context bound through three training configurations (Figure 3). All variants use graph context and share the encoder, auxiliary head, finite-margin objective, and training settings. We evaluate all three cross-KG datasets and WebQSP/CWQ in-domain (Appendix E.4).

Contribution to transfer. Reference supervision improves cross-KG retrieval and QA on all three datasets. The context bound adds further gains. Source-answer ranking trains semantic relevance. The bound limits contextual changes to this reference. The consistent gains support learning semantic relevance and constraining contextual adjustments together. Paired 95% intervals for their combined QA gains lie above zero on all three targets (Appendix E.2). At a fixed sample size, multi-source training also yields modest MRR gains (Appendix E.6).

Role of context. Role of context. At inference, the jointly trained reference S_0 exceeds S_θ in cross-KG H@1 and QA F1 on WebQSP and CWQ (Appendix E.5). Context improves in-domain H@1 on both datasets. The in-domain QA differences are small. Both paired intervals include zero. These comparisons separate training benefits from contextual scoring at inference. All main results use S_θ.

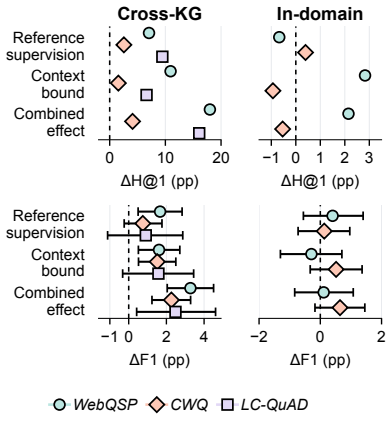

Figure 3: Ablations of reference supervision and the context bound. Error bars denote paired 95% bootstrap intervals for QA F1 gains.

4.4 GENERALIZATION ACROSS QUERY DISTANCES

We examine whether the cross-KG gains extend to longer query-graph paths. We define query distance as the shortest path from an entity constant to the answer variable in the query graph. This measure captures path length. Questions in either group can contain additional constraints. Appendix E.2 gives the group statistics. KGSEEK improves MRR and QA F1 across all six dataset–distance groups (Figure 4). It retains these gains on queries that require at least two relation steps to reach the answer variable. Both readers benefit in these groups. The transfer advantage therefore extends beyond queries of distance one.

4.5 IN-DOMAIN PERFORMANCE

The transfer-oriented design also supports in-domain learning. KGSEEK achieves the highest H@1 and MRR on all three datasets (Table 2). Its CWQ H@1 is close to KGFR (Cui et al., 2026). For downstream QA, KGSEEK achieves the best F1 on CWQ and LC-QuAD with both readers. Its WebQSP F1 is within 0.88 points of the strongest baseline (Table 3). Together with the scoring comparison above, these results show that the same architecture can use target supervision to refine candidate rankings while remaining effective for QA.

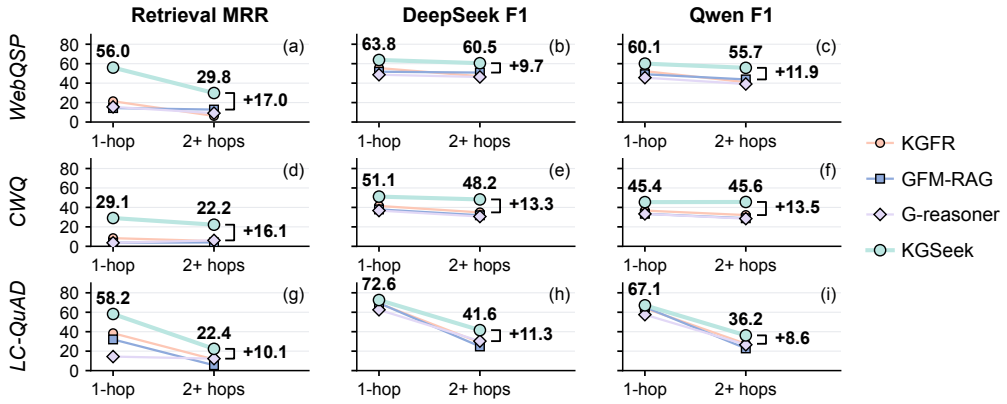

Figure 4: Cross-KG retrieval and QA by query distance (%). Labels give KGSEEK scores; brackets show gains over the strongest baseline on queries of distance ≥ 2 (pp).

4.6 CASE STUDY

Figure 5 shows semantic matching and contextual refinement on WebQSP with DeepSeek. The cross-KG example asks where *Benjamin Franklin* studied. KGFR's reader names an institution he founded, whereas KGSEEK ranks *Boston Latin School* first and supports the correct answer. The variant trained without the context bound ranks the school tenth. This example highlights confusion between education and founding an institution. The in-domain question asks what type of cancer *Eva Perón* had. Contextual scoring moves *Cervical cancer* from second to first, above *Cancer*. All compared configurations yield the correct reader answer. This shows how context can distinguish a specific answer from a broader category under the same relation. Appendix E.7 gives selection criteria and supporting evidence.

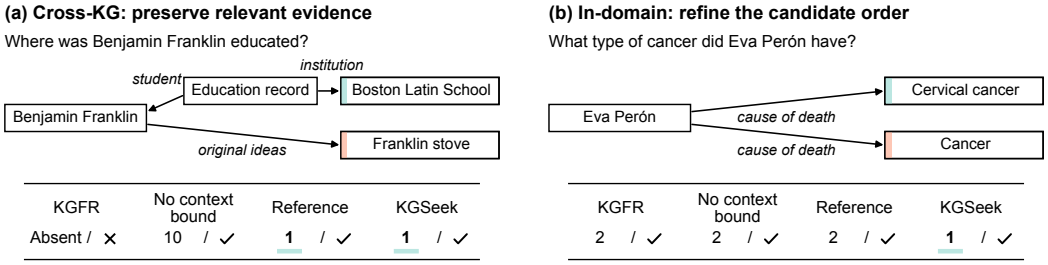

Figure 5: Ranking examples. Teal marks gold answers. Salmon marks competing candidates. Cells show gold-answer rank in reader input and QA correctness (✓/×). Edges show the KG evidence.

5 CONCLUSION

We presented KGSEEK, a multi-source foundation retriever for multi-hop reasoning across KGs. It learns semantic relevance from source QA labels and uses bounded graph context to refine answer rankings. Maximum-mean path scoring preserves the adjustment bound across path lengths. A single source-trained model improves retrieval on unseen KGs and downstream QA with two LLM readers. The same architecture improves in-domain retrieval and maintains competitive QA performance. These findings highlight the value of learning transferable semantic relevance while controlling how graph context changes answer rankings. Future work can extend retrieval to more KGs and incorporate logical and numerical reasoning to further improve performance.

AI USE STATEMENT

We used generative AI tools to assist with data preprocessing, baseline implementation, and manuscript polish. The authors take responsibility for the final content.

ETHICS STATEMENT

We use public KGQA benchmarks for computational experiments. KGs and pretrained language models can contain incomplete, outdated, or biased information. Better retrieval does not guarantee factual correctness or fair coverage. Applications should respect data licenses and privacy, with human review for high-stakes decisions.

REPRODUCIBILITY STATEMENT

The source code accompanies the appendix. Section 3 describes the model and training objectives. Appendix B provides architecture details, the path-scoring algorithm, and proofs. Appendix C reports baseline adaptations, hyperparameters, and checkpoint selection. Appendix D describes data preprocessing and graph construction. Appendix F specifies QA readers, prompts, and metrics.

REFERENCES

Arvindh Arun, Sumit Kumar, Mojtaba Nayyeri, Bo Xiong, Ponnurangam Kumaraguru, Antonio Vergari, and Steffen Staab. SEMMA: A Semantic Aware Knowledge Graph Foundation Model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 31825–31848, 2025. doi: 10.18653/v1/2025.emnlp-main.1621.

Beijing Academy of Artificial Intelligence. BGE-large-en-v1.5. Model release, 2023. URL `https://huggingface.co/BAAI/bge-large-en-v1.5`.

Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyiu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. KQA Pro: A Dataset with Explicit Compositional Programs for Complex Question Answering over Knowledge Base. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6101–6119, 2022. doi: 10.18653/v1/2022.acl-long.422.

Yuanning Cui, Zequn Sun, and Wei Hu. A Prompt-Based Knowledge Graph Foundation Model for Universal In-Context Reasoning. In *Advances in Neural Information Processing Systems*, pp. 7095–7124, 2024. doi: 10.52202/079017-0227.

Yuanning Cui, Zequn Sun, and Wei Hu. Transfer-and-Fusion: Integrated Link Prediction Across Knowledge Graphs. *IEEE Transactions on Knowledge and Data Engineering*, 37(5):3062–3074, 2025. doi: 10.1109/TKDE.2025.3544255.

Yuanning Cui, Zequn Sun, Wei Hu, and Zhangjie Fu. KGFR: A Foundation Retriever for Generalized Knowledge Graph Question Answering. arXiv preprint arXiv:2511.04093, 2026. URL `https://arxiv.org/abs/2511.04093v2`.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards Foundation Models for Knowledge Graph Reasoning. In *International Conference on Learning Representations*, pp. 31598–31619, 2024.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. In *Advances in Neural Information Processing Systems*, pp. 132876–132907, 2024. doi: 10.52202/079017-4224.

Xingyue Huang, Pablo Barcelo, Michael M. Bronstein, Ismail Ilkan Ceylan, Mikhail Galkin, Juan L Reutter, and Miguel Romero Orth. How Expressive are Knowledge Graph Foundation Models? In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 25021–25058, 2025.

Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. UniKGQA: Unified Retrieval and Reasoning for Solving Multi-hop Question Answering Over Knowledge Graph. In *International Conference on Learning Representations*, 2023.

Jinwoo Kim, Xingyue Huang, Krzysztof Olejniczak, Kyungbin Min, Michael Bronstein, Seunghoon Hong, and Ismail I Ceylan. Flock: A Knowledge Graph Foundation Model via Learning on Random Walks. In *International Conference on Learning Representations*, pp. 36664–36705, 2026.

Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang. InGram: Inductive Knowledge Graph Embedding via Relation Graphs. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 18796–18809, 2023.

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*, pp. 6061–6089, 2025.

Junnan Liu, Qianren Mao, Weifeng Jiang, and Jianxin Li. KnowFormer: Revisiting Transformers for Knowledge Graph Reasoning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 31669–31690, 2024.

Linhao Luo, Yuan-Fang Li, Reza Haffari, and Shirui Pan. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In *International Conference on Learning Representations*, pp. 14400–14423, 2024.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuan-Fang Li, Chen Gong, and Shirui Pan. Graph-constrained Reasoning: Faithful Reasoning on Knowledge Graphs with Large Language Models. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 41540–41565, 2025a.

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*, pp. 36371–36405, 2025b. doi: 10.52202/085713-1222.

Linhao Luo, Zicheng Zhao, Junnan Liu, Zhangchi Qiu, Junnan Dong, Serge Panev, Chen Gong, Thuy-Trang Vu, Gholamreza Haffari, Dinh Phung, Alan Liew, and Shirui Pan. G-reasoner: Foundation Models for Unified Reasoning over Graph-structured Knowledge. In *International Conference on Learning Representations*, pp. 63769–63793, 2026.

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. doi: 10.18653/v1/2025.findings-acl.856.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. In *Advances in Neural Information Processing Systems*, pp. 14501–14515, 2022. doi: 10.52202/068431-1054.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019. doi: 10.18653/v1/D19-1410.

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. doi: 10.18653/v1/2021.emnlp-main.341.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William 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. doi: 10.18653/v1/D18-1455.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. PullNet: Open Domain Question Answering with Iterative Retrieval on Knowledge Bases and Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2380–2390, 2019. doi: 10.18653/v1/D19-1242.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *International Conference on Learning Representations*, pp. 3868–3898, 2024.

Zequn Sun, Jiacheng Huang, Jinghao Lin, Xiaozhou Xu, Qijin Chen, and Wei Hu. Joint Pre-training and Local Re-training: Transferable Representation Learning on Multi-source Knowledge Graphs. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2132–2144, 2023. doi: 10.1145/3580305.3599397.

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: Human Language Technologies, Volume 1 (Long Papers)*, pp. 641–651, 2018. doi: 10.18653/v1/N18-1059.

Komal Teru, Etienne Denis, and Will Hamilton. Inductive Relation Prediction by Subgraph Reasoning. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 9448–9457, 2020.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. LC-QuAD: A Corpus for Complex Question Answering over Knowledge Graphs. In *The Semantic Web – ISWC 2017, Part II*, pp. 210–218, 2017. doi: 10.1007/978-3-319-68204-4_22.

Tianjun Yao, Haoxuan Li, Zhiqiang Shen, Pan Li, Tongliang Liu, and Kun Zhang. Learning Efficient and Generalizable Graph Retriever for Knowledge-Graph Question Answering. arXiv preprint arXiv:2506.09645, 2025. URL `https://arxiv.org/abs/2506.09645`.

Wen-tau Yih, Matthew Richardson, Chris 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 (Volume 2: Short Papers)*, pp. 201–206, 2016. doi: 10.18653/v1/P16-2033.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do Transformers Really Perform Badly for Graph Representation? In *Advances in Neural Information Processing Systems*, pp. 28877–28888, 2021.

Junchi Yu, Yujie Liu, Jindong Gu, Philip Torr, and Dongzhan Zhou. Can Knowledge-Graph-based Retrieval Augmented Generation Really Retrieve What You Need? In *Advances in Neural Information Processing Systems*, pp. 95653–95682, 2025. doi: 10.52202/085713-3200.

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 (Volume 1: Long Papers)*, pp. 5773–5784, 2022. doi: 10.18653/v1/2022.acl-long.396.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. Variational Reasoning for Question Answering With Knowledge Graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018. doi: 10.1609/aaai.v32i1.12057.

Rongzhi Zhu, Xiangyu Liu, Zequn Sun, Yiwei Wang, and Wei Hu. Mitigating Lost-in-Retrieval Problems in Retrieval Augmented Multi-Hop Question Answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22362–22375, 2025. doi: 10.18653/v1/2025.acl-long.1089.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. In *Advances in Neural Information Processing Systems*, pp. 29476–29490, 2021.

A Cross-KG Transfer Analysis

We evaluate KGFR transfer on validation questions from three datasets and two KG families (Figure 1). Table 4 reports all nine source–target pairs, with training sources in rows and target questions and graphs in columns. The diagonal gives in-domain references with matched training sizes.

Table 4: KGFR cross-dataset transfer: validation Hits@1 / MRR (%). Bold: in-domain results.

Source \ target	KQA Pro	WebQSP	CWQ
KQA Pro	**44.83 / 55.49**	6.09 / 15.37	3.11 / 7.96
WebQSP	13.22 / 21.69	**61.74 / 67.82**	15.20 / 25.84
CWQ	4.60 / 16.59	20.43 / 31.61	**16.93 / 26.37**
Target questions	174	230	1,382
Graph answer coverage	100%	100%	100%
APP answer reachability	92.53%	96.52%	90.52%

Question subset construction. For KQA Pro, we select entity-path programs with one or two `Relate` operations. For WebQSP and CWQ, we select questions with one topic entity, an entity answer, and a directed topic-to-answer shortest distance of one or two in the released question-local graph. These criteria match path structure while allowing different semantic operators. Each source contains 823 one-hop and 177 two-hop examples. With seed 123, we split these 1,000 examples into 900 training and 100 validation examples. We select checkpoints on the source validation split. WebQSP/CWQ share a union graph of 8,309,195 triples.

Interpretation. For target t, retention is the ratio $\mathrm{Retain}(s\to t) = \mathrm{H@1}(s\to t)/\mathrm{H@1}(t\to t)$. We define MRR retention analogously. KQA Pro-to-WebQSP/CWQ retains 22.7%/30.2% of in-domain MRR. Compared with training on the other Freebase dataset, the KQA Pro source lowers Hits@1 by 14.35/12.08 points on WebQSP/CWQ. Same-KG transfer is asymmetric. CWQ-to-WebQSP retains 33.10% of the WebQSP in-domain score. WebQSP-to-CWQ retains 89.74% of the CWQ in-domain score. KQA Pro-to-Freebase retains 9.86% and 18.38% on WebQSP and CWQ, respectively. Reverse cross-KG transfer is also weak. The same-KG control covers Freebase, which has two QA datasets. The comparisons vary source question distributions and KG schemas together. Fixed target coverage and reachability isolate ranking differences within the same candidate support.

B Model Architecture and Theoretical Analysis

B.1 Question-Conditioned Graph Encoding

The sparse Graph Transformer uses the same fixed support in all layers. Node states start from $\mathbf{h}_v^{(0)} = \mathbf{t}_v$. Let $\mathbf{q}$ denote the raw frozen question embedding, before the unit normalization used to obtain $\bar{\mathbf{q}}$. At layer ℓ, a question-aware transformation updates relation states $\boldsymbol{\rho}_r^{(\ell)} = \mathrm{LN}(F_\ell([\mathbf{q}; \boldsymbol{\rho}_r^{(\ell-1)}]))$, where F_ℓ is an affine map. Real KG relations start from frozen embeddings of the backbone's relation-name strings. The artificial self-loop uses a learned embedding. In the attention equations below, node states come from layer $\ell-1$ and relation states from layer ℓ. We omit layer indices for readability. Let $\mathcal{I}_v^{\mathrm{enc}} = \{(u,r) : (u,r,v) \in \mathcal{E}_q \cup \mathcal{E}_{\mathrm{self}}\}$ collect incoming edges, where $\mathcal{E}_{\mathrm{self}}$ contains the artificial encoder self loops. For an attention head of dimension d_h and incoming edge (u,r,v), the update uses

$$\mathbf{Q}_v = W_Q W_0 \mathbf{h}_v + W_q \mathbf{q}, \qquad \mathbf{K}_{ur} = W_K \mathbf{h}_u + W_{rK} \boldsymbol{\rho}_r,$$
$$a_{urv} = \mathbf{Q}_v^\top \mathbf{K}_{ur} / \sqrt{d_h} + b_r(\boldsymbol{\rho}_r), \qquad \alpha_{urv} = \frac{\exp(a_{urv})}{\sum_{(u',r')\in\mathcal{I}_v^{\mathrm{enc}}} \exp(a_{u'r'v})}. \tag{7}$$

Here b_r is a shared linear function of relation features. Values are $W_V \mathbf{h}_u + W_{rV} \boldsymbol{\rho}_r$. The encoder concatenates and projects the incoming sums from all heads. It adds the projected input $W_0 \mathbf{h}_v$ and applies layer normalization. A two-layer feed-forward network (FFN) uses ReLU and a second post-normalized residual update. Dropout applies to attention weights, hidden feed-forward activations,

and residual branch outputs. Each layer shares its own parameters across KGs. Attention operates on support edges. The semantic reference applies the shared edge head directly to fixed topic roles $\mathbf{t}_v$.

B.2 Shared Semantic and Contextual Scoring

For endpoint states $\mathbf{x}_u, \mathbf{x}_v$, the edge head combines four projected inputs:

$$\begin{aligned} \boldsymbol{\phi} &= \tfrac{1}{2}\left[\mathrm{LN}_0(P_u\mathbf{x}_u) + \mathrm{LN}_0(P_v\mathbf{x}_v) + \mathrm{LN}_0(P_r\bar{\mathbf{r}}) + \mathrm{LN}_0(P_q\bar{\mathbf{q}})\right], \\ f_\psi(\mathbf{x}_u, \mathbf{x}_v, q, r) &= \mathbf{a}^\top \mathrm{GELU}(\boldsymbol{\phi}) + b. \end{aligned} \tag{8}$$

All projections are bias-free and have output dimension 64. LN_0 uses no learned affine parameters. We initialize the output layer at $\mathbf{a} = \mathbf{0}, b = 0$. Reference and contextual scores share all head parameters ψ and differ only in their endpoint inputs. The full parameter set θ also includes the graph encoder and auxiliary entity head. The contextual score requires one graph encoding. The reference applies the head directly to fixed topic roles. Source answers train S_0, while A remains frozen.

B.3 Path Aggregation and Computational Complexity

We compute the maximum average walk score through maximum-sum prefixes:

$$\begin{aligned} D_0(v) &= \begin{cases} 0 & v \in T, \\ -\infty & v \notin T, \end{cases} \qquad \mathcal{R}(w; q, v) = \max_{1 \le h \le H} D_h(v)/h, \\ D_h(v) &= \max_{(u,r,v)\in\mathcal{E}_q} [D_{h-1}(u) + w(q, u, r, v)]. \end{aligned} \tag{9}$$

We prove by induction on h that $D_h(v)$ in Eq. (9) is the maximum sum over all exactly-h walks from T to v. The base condition represents zero-length prefixes at topic entities. Every h-edge walk ends in one incoming edge (u, r, v). Maximizing its preceding sum yields the recurrence. We divide by h and maximize over allowed lengths to obtain Eq. (4) exactly. We remove artificial encoder self loops before the recurrence and retain self-relations from the KG.

The readout costs $O(H(|\mathcal{V}_q| + |\mathcal{E}_q|))$ scalar operations and $O(|\mathcal{V}_q|)$ state memory for forward inference, plus stored edges and weights. Encoder projection and attention add cost. Let R_q denote the relation types in the support graph. With dense feature projections, one sparse Transformer layer has a term of order $O((|\mathcal{V}_q| + |R_q|)d^2 + |\mathcal{E}_q|d)$. Training also stores states for backpropagation.

B.4 Proof of Proposition 1

For any allowed walk p, write its frozen mean as a_p and its final mean as s_p. Define $\delta_\theta(q, a) = w_\theta(q, a) - c(q, r_a)$, where r_a is the relation of edge a. Since every correction lies in $[-B, B]$,

$$|s_p - a_p| = \left|\frac{1}{|p|}\sum_{a\in p}\delta_\theta(q, a)\right| \le \frac{1}{|p|}\sum_{a\in p}|\delta_\theta(q, a)| \le B. \tag{10}$$

The set of walks is finite because the graph and horizon are finite, even though vertices may repeat. More generally, for two edge-weight functions w, w' on the same support, each walk mean changes by at most $\|w - w'\|_\infty$. Maximizing over a common walk set preserves this bound:

$$\|\mathcal{R}(w) - \mathcal{R}(w')\|_\infty \le \|w - w'\|_\infty, \tag{11}$$

where the norm on the left ranges over reachable entities. For a fixed reachable endpoint, each $s_p \le a_p + B \le \max_{p'} a_{p'} + B$. Thus, $S_\theta(e) \le A(e) + B$. The symmetric inequality gives $S_\theta(e) \ge A(e) - B$. For two reachable endpoints e, f,

$$|[S_\theta(e) - S_\theta(f)] - [A(e) - A(f)]| \le |S_\theta(e) - A(e)| + |S_\theta(f) - A(f)| \le 2B. \tag{12}$$

If $A(e) - A(f) > 2B$, then $S_\theta(e) - S_\theta(f) > 0$. The final ranking therefore preserves a sufficiently large frozen-score gap. The proof uses the same support and walk set for S_θ and A, as defined in Eq. (4). The same argument for $w_0 - c = B\tanh z_0$ gives $|S_0 - A| \le B$.

For the contextual adjustment, $\tanh$ is 1-Lipschitz and $|\Delta| \le \kappa$. On every real edge,

$$|w_\theta - w_0| = B|\tanh(z_0 + \Delta) - \tanh z_0| \le B|\Delta| \le B\kappa. \tag{13}$$

Averaging over a walk and maximizing over the same walk set therefore gives $|S_\theta - S_0| \le B\kappa$. Applying the triangle inequality to two candidate scores bounds the pairwise change by $2B\kappa$. Consequently, a reference gap above $2B\kappa$ remains positive. These statements also hold at $\kappa = 0$ with $\Delta = 0$. Both residuals are bounded by B. This gives the tighter score bound $B\min(\kappa, 2)$.

B.5 Auxiliary Entity Supervision

Let $\mathcal{C}_q^{\mathrm{aux}}$ be the encoder's local node set after the same answer-eligibility mask used for retrieval. Unlike $\mathcal{C}_q$, it is not restricted to nodes with a finite score from a real-edge walk of length $1, \dots, H$. We train a bias-free linear entity head $g_\theta(q, e) = \mathbf{a}_{\mathrm{aux}}^\top \mathbf{h}_e^{(L)}$ with the listwise loss

$$\mathcal{L}_{\mathrm{aux}} = \log \sum_{e \in \mathcal{C}_q^{\mathrm{aux}}} \exp g_\theta(q, e) - \log \sum_{y \in Y \cap \mathcal{C}_q^{\mathrm{aux}}} \exp g_\theta(q, y). \tag{14}$$

We skip a training query if no eligible gold answer has a finite path score. For retained queries, the auxiliary set contains that gold answer and may include additional encoder nodes. An empty set of ranking negatives gives zero ranking loss but still permits auxiliary supervision. The deployed path score excludes the auxiliary score.

B.6 Gradient Analysis of the Finite-Margin Objective

For either learned ranking $F \in \{S_\theta, S_0\}$ and a selected negative n, let $g_n = F(q, y_F^*) - F(q, n)$. Backpropagation treats both selections as fixed. The ranking term is $[(m - g_n)_+/\tau]^2$. Its derivative with respect to g_n is $-2(m - g_n)/\tau^2$ when $g_n < m$. Otherwise, it is zero. Thus, a comparison that meets the margin contributes zero ranking gradient. Other comparisons and the auxiliary objective can still update the shared parameters. We reselect hard negatives as training proceeds.

C Baseline Adaptations and Experimental Settings

Baseline configurations. KGFR uses a three-layer progressive GNN with 1,024-dimensional hidden states. It retains its multi-positive listwise entity-ranking objective and uses the frozen text encoder and APP settings in Section 4.1.

GFM-RAG uses a six-layer, 512-dimensional reasoner with frozen BGE question and relation embeddings. We pretrain it on source KG completion for 10 epochs with 256 batches per source per epoch, batch size 8, and 256 strict negatives. We replace head prediction with inverse-relation tail prediction and compute loss from sampled entity scores. Source QA training combines balanced binary cross-entropy (BCE) and listwise cross-entropy with weights 0.3 and 0.7.

G-reasoner uses a six-layer, 1,024-dimensional graph reasoning network with early/late entity fusion and frozen BGE surface-name embeddings. We train it from scratch on source QA data. Following its released entity-supervised configuration, we combine balanced BCE, listwise cross-entropy, and normalized mean-squared-error distillation with weights 0.3, 0.7, and 1.

For QA training, both models use AdamW at 5×10^{-4}, one question per update, gradient clipping at 5, and a 30-epoch maximum. The selection rule below chooses cross-KG epochs 25 and 23, respectively. We adapt both document-ranking pipelines to rank entities on the shared APP graph and supply evidence to the common reader. For G-reasoner, BGE replaces the Qwen text interface. Our graph contains entities and relations in place of document and community nodes.

In-domain G-reasoner retains these settings and uses each dataset's full training and validation splits (Table 1) and background graph. Validation selects epochs 26, 20, and 27 on WebQSP, CWQ, and LC-QuAD. Retrieval and both readers use the same selected checkpoint.

In-domain GFM-RAG pretraining. We pretrain on a compact subgraph of each Freebase or DBpedia training background. With seed 123, we sample at least one fact per relation and expand by bounded multi-seed breadth-first search. The subgraph contains 100,000 nodes and at most 250,000 forward facts, or 500,000 directed edges with inverses. Sampling depends on KG facts, independently of QA labels. We pretrain for 10 epochs with batch size 2, 256 batches per epoch, and 256 negatives. QA training uses every training question and the original APP supports. Validation

selects epochs 11, 19, and 15 on WebQSP, CWQ, and LC-QuAD. WebQSP/CWQ test-answer reachability is 93.12%/85.70%. The LC-QuAD pretraining subgraph contains 250,000 forward facts across all 1,747 relations with facts. Its QA splits and evaluation graph follow Appendix D.1.

Model configuration. The sparse Transformer and edge head have 44,359,745 trainable parameters (Table 5). The 64-dimensional head contributes 262,209 parameters; the text encoder remains frozen. The question conditions attention queries. Each relation update uses the raw question embedding and previous relation state. The shared edge head produces semantic and contextual scores for path-based entity ranking. For the reference, we separately unit-normalize and detach question and directed-relation embeddings. Relation text specifies inverse traversal. The encoder learns self-loop relations for encoding only.

Support construction. At each step, KGFR's APP rule (Cui et al., 2026) examines outgoing $(\mathrm{head}, \mathrm{relation})$ groups of all entities reached before that step. Groups with at most λ edges retain every edge. Larger groups retain only edges whose tails were already reached. We add new tails after filtering. All encoder layers reuse the complete, deduplicated union of retained real edges over H steps. We choose encoder depth L independently of retrieval horizon H.

Implementation. We use topology caching, TF32 execution, and size-dependent activation recomputation to accelerate training. They preserve the support graph and scoring functions.

Answer candidates. The answer mask excludes nodes whose stored names begin with the Freebase identifier prefixes `m.` or `g.`. These nodes remain in the support graph as possible intermediate nodes. Ranking candidates also require a finite score from a real-edge walk of length $1, \ldots, H$. The auxiliary set includes all eligible encoder nodes (Appendix B.5).

Schedules and checkpoint selection. We use seed 123 and select cross-KG checkpoints by source-validation macro Hits@1, with MRR as the tie-breaker. Each selected model serves all three targets. In-domain selection uses target validation. WebQSP/CWQ validation and earlier test analyses informed method development. LC-QuAD is held out of development and cross-KG model selection.

QA training uses early stopping with a 20-epoch minimum and patience five for KGSEEK, its two ablations, GFM-RAG, G-reasoner, and cross-KG KGFR. KGSEEK and its ablations train for at most 30 epochs on the multi-source, WebQSP, and LC-QuAD tasks, and 60 on CWQ. We keep the reference weight and context bound fixed across tasks. KGFR retains its listwise objective and uses learning rate 10^{-4} for cross-KG training. On LC-QuAD, KGFR and KGSEEK train from scratch with the same 30-epoch schedule and stopping rule. Their learning rates are 10^{-4} and 5×10^{-5}, respectively. Table 6 lists validation-selected epochs. Budgets and hyperparameters remain method-specific.

Retrieval metrics. Hits@k measures whether at least one gold entity appears in the top k. Recall@20 averages the fraction of each question's full gold answer set retrieved, including answers absent from the candidate graph. MRR uses the first correct rank among all finite-score candidates, without truncation at the top-20 reader budget. Unreachable questions contribute zero, and global entity IDs break score ties deterministically. Retrieval and QA use the same checkpoint for each condition. We report one training seed and one response per question, retriever, and reader. Appendix F specifies readers and scoring; Appendix E.2 describes paired intervals.

D Dataset Construction and Supervision

KQA Pro source. We retain the entity-centric, single-chain program family

$$\texttt{Find} \rightarrow (\texttt{Relate} \rightarrow \texttt{FilterConcept})^{1:3} \rightarrow \texttt{What}.$$

We re-execute programs on the released Wikidata graph and preserve concept filters as type restrictions. Each retained example has one resolved entity answer. The filter excludes literal/numeric outputs, counting, comparisons, qualifiers, and branching. We retain 1,364 training examples (1,123 one-hop, 241 two-hop, no three-hop) and 174 validation examples, together with the source graph.

Table 5: KGSEEK hyperparameters shared across in-domain and cross-KG settings.

Setting	Value	Setting	Value
Text encoder	BGE-large-en-v1.5	Text encoder updates	Frozen
State dimension d	1,024	Transformer layers L	3
Attention heads	8	FFN width multiplier	2
Residual dropout	0.3	Attention dropout	0.1
APP threshold λ	10	Walk horizon H	3
Edge projection width	64	Correction bound B	0.3
Ranking margin m	0.05	Ranking-loss scale τ	0.1
Hard negatives K	20 per score	Auxiliary weight η	0.5
Reference weight β	0.1	Context bound κ	1
Optimizer	Adam	Training seed	123
Learning rate	5×10^{-5}	Weight decay	10^{-4}
Batch size	1 question	Gradient clipping	5
Precision mode	TF32 enabled	Inference score	S_θ

Table 6: Epochs selected by validation performance.

Setting	Source	KGFR epoch	KGSEEK epoch
In-domain	WebQSP	11	27
In-domain	CWQ	42	25
In-domain	LC-QuAD	25	18
Cross-KG	Balanced KQA Pro + MetaQA	16	7

MetaQA source. We sample the vanilla two-hop WikiMovies split to match KQA Pro's training and validation sizes. The original pool contains 118,980 training and 14,872 development examples. Movie-domain relations complement KQA Pro's broader topics, while templated questions add limited linguistic diversity.

Training supervision. Both sources provide (G, q, T, Y) examples. KQA Pro programs select the subset; question, topic, and answer labels supervise training. Round-robin sampling alternates between equally sized sources and preserves their hop distributions. Queries without an eligible reachable answer contribute zero loss to Eq. (1). We skip them without replacement, so update counts can differ across sources. Each ranking selects its highest-scoring gold answer and hard negatives outside the annotated answer set. At inference, we score every eligible entity.

Names, identifiers, and target access. Entity IDs index structure and evaluation labels. The model uses fixed topic-role features instead of learned entity embeddings; readers receive readable names. The backbone keeps KGFR-style relation names. The semantic readout separates punctuation and word boundaries and specifies traversal direction. At deployment, the fixed model uses target structure and relation labels without target QA demonstrations or fine-tuning. We build Freebase backgrounds by deduplicating the public graph fields across training, validation, and test splits, independently of answer columns. Cross-KG evaluation combines WebQSP and CWQ facts; each in-domain experiment uses its own dataset's union. Transfer concerns QA supervision: the pretrained text encoder may have encountered target-KG facts.

D.1 LC-QUAD DATA AND GRAPH CONSTRUCTION

Entity-answer subset. We execute the official LC-QuAD 1.0 test queries (Trivedi et al., 2017) on fixed English DBpedia 2016-04 components. Of 1,000 queries, 83 use ASK, 123 use COUNT, and 794 use SELECT with basic graph patterns. We retain 740 SELECT queries with nonempty DBpedia-resource answer sets. The remaining 54 yield mixed literal/entity answers (49), external URIs (four), or empty answers (one). Selection depends on query and answer types, independently of predictions or APP reachability. We preserve conjunctions and type constraints. The nearest

topic-to-answer query distance is one for 487 queries and two for 253. We fix cross-KG models and hyperparameters before this evaluation.

In-domain training subset. We apply the same filter to the official training split. Removing one test-question duplicate leaves 2,958 examples. A grouped, deduplicated split with seed 123 assigns 2,662 to training and 296 to validation. Test text serves only to remove overlap. The training graph preserves their APP supports under the same snapshot and settings. It contains 1,154,855 nodes and 6,111,363 forward facts, separate from the fixed 740-question test graph. All four methods share these assets for retrieval and QA with both readers.

Graph and topic entities. The snapshot includes mapping-based objects and literals, infobox properties, instance types and transitive types, specific mapping-based properties, and English labels.[1] Retrieval uses IRI-to-IRI edges with ontology/property relations or `rdf:type`. Class nodes remain eligible candidates. We derive text from fixed URI names with consistent word-boundary rules. SPARQL resource constants supply oracle topic entities. We use the queries offline for dataset selection and gold answers. Retrieval and support construction use the supplied topics, independently of SPARQL text, gold relations, and gold paths.

Shared support and evaluation. All methods use APP with $H = 3$ and $\lambda = 10$ and compute degrees on the fixed background. We verify the compact graph's APP nodes and edges against the full snapshot at every hop for all 740 questions. At least one gold answer is reachable for 707 questions (95.54%); all are reachable for 634 (85.68%). We include all 33 unreachable questions and use complete gold answer sets, including graph-external answers, in recall. Each cross-KG model reuses its Freebase checkpoint without target-based training or selection. This target tests transfer across KG representations and schemas, allowing DBpedia and source KGs to share entities and facts.

E Supplementary Experiments and Analysis

E.1 Complete QA Results

Tables 7 and 8 add Hit and EM to Table 3 using the same checkpoints and reader outputs. Bold marks the best score per dataset and setting.

E.2 Statistical Analysis

We estimate pointwise paired 95% percentile intervals from 10,000 bootstrap draws with seed 123. Each draw resamples the same questions with replacement for both methods within a dataset. All test questions contribute, including invalid outputs and refusals. The intervals measure question-sampling variation for fixed checkpoints and reader responses.

Cross-KG performance. All 12 intervals favor KGSeek over KGFR, the strongest cross-KG baseline on these metrics (Table 9). They cover all 1,628 WebQSP, 3,531 CWQ, and 740 LC-QuAD questions and support retrieval gains on both KGs and QA gains with both readers. LC-QuAD's lower F1 bounds are 4.05 points for DeepSeek and 2.18 for Qwen.

Query-distance groups. We parse original queries and match them to test predictions by question ID and text. Each predicate connects two query nodes. We compute the shortest undirected distance from any entity constant to the answer variable, counting each Freebase mediator edge as one hop. We preserve subquery variable scope and separate UNION branches. We minimize distance across branches and valid parses. Literal constraints and filters add no entity-to-answer paths. This measure includes paths from constraint entities and describes query structure, not full logical complexity.

WebQSP has 1,109 one-hop and 518 longer-distance questions (517 at distance two, one at three). One alias-only query lacks an entity constant and distance label but remains in the main results. Fifteen questions have both one-hop and longer-distance parses. The minimum-distance rule assigns them to the one-hop group. CWQ has 1,551 one-hop and 1,980 longer-distance questions (1,431

[1] Official release: `https://downloads.dbpedia.org/2016-04/core-i18n/en/`.

Table 7: QA with DeepSeek V4 Flash (%). Bold: best per setting.

Setting	Retriever	WebQSP			CWQ			LC-QuAD		
		Hit	F1	EM	Hit	F1	EM	Hit	F1	EM
Control	No retrieval	57.13	40.60	23.96	37.13	31.35	25.57	41.76	32.21	26.08
Cross-KG	KGFR	66.52	52.79	38.02	43.30	37.80	31.80	65.81	55.35	44.73
	GFM-RAG†	65.97	51.42	35.26	39.45	34.58	29.28	65.27	54.20	43.92
	G-reasoner†	63.76	47.87	32.43	38.91	33.32	27.64	60.68	51.51	41.49
	KGSEEK	**80.16**	**62.73**	**45.02**	**56.05**	**49.49**	**41.66**	**75.00**	**61.98**	**49.46**
In-domain	KGFR	84.09	66.72	47.91	62.79	54.25	44.58	77.84	64.22	52.16
	GFM-RAG†	**85.07**	**68.47**	**49.88**	58.99	51.56	42.99	77.57	64.63	51.89
	G-reasoner†	84.95	**68.47**	49.20	60.78	52.38	42.85	78.92	65.84	53.78
	KGSEEK	84.77	67.59	48.46	**63.98**	**55.69**	**46.16**	**80.95**	**66.74**	**54.86**

Table 8: QA with Qwen3.6-27B (%). Bold: best per setting.

Setting	Retriever	WebQSP			CWQ			LC-QuAD		
		Hit	F1	EM	Hit	F1	EM	Hit	F1	EM
Control	No retrieval	49.51	33.74	20.15	27.30	22.96	19.23	31.89	22.36	17.43
Cross-KG	KGFR	62.16	48.67	34.34	39.90	34.15	28.35	61.49	51.62	41.08
	GFM-RAG†	62.16	47.41	32.74	36.17	30.90	25.66	60.95	50.81	41.35
	G-reasoner†	58.48	43.52	28.75	36.45	30.61	24.84	56.89	46.94	36.08
	KGSEEK	**75.25**	**58.65**	**40.79**	**51.88**	**45.53**	**37.98**	**68.38**	**56.55**	**45.81**
In-domain	KGFR	77.83	60.87	41.83	57.09	49.46	40.61	70.68	57.93	46.08
	GFM-RAG†	78.56	**62.16**	43.00	54.43	47.28	39.37	70.68	58.22	46.22
	G-reasoner†	78.38	62.06	43.12	55.68	47.66	38.60	71.62	59.32	47.84
	KGSEEK	**78.69**	61.98	**43.43**	**58.82**	**51.26**	**42.51**	**74.73**	**61.77**	**50.54**

Table 9: Cross-KG gains over KGFR (pp). Gains appear above paired 95% bootstrap intervals.

	Retrieval		QA F1	
Dataset	Hits@1	MRR	DeepSeek V4 Flash	Qwen3.6-27B
WebQSP	+27.83 [25.31, 30.47]	+31.08 [28.86, 33.33]	+9.94 [8.19, 11.69]	+9.98 [8.29, 11.76]
CWQ	+8.98 [7.84, 10.14]	+18.52 [17.43, 19.64]	+11.69 [10.37, 13.01]	+11.38 [10.10, 12.69]
LC-QuAD	+14.46 [10.95, 18.11]	+16.74 [13.80, 19.74]	+6.63 [4.05, 9.28]	+4.93 [2.18, 7.70]

at two, 421 at three, 128 at four). All 740 LC-QuAD questions have labels: 487 one-hop and 253 two-hop. Group denominators include unreachable answers and invalid outputs. Weighted scores reproduce the main results when they include WebQSP's unlabeled question.

We use 10,000 paired bootstrap draws per group (seed 123). In the longer-distance groups, MRR gains over KGFR are 23.56 points on WebQSP (95% interval: [20.38, 26.75]), 16.82 on CWQ ([15.42, 18.21]), and 10.90 on LC-QuAD ([7.21, 14.64]). DeepSeek F1 gains are 13.42 ([10.28, 16.50]), 13.32 ([11.61, 15.08]), and 12.65 ([8.37, 17.13]) points, respectively. Qwen F1 gains are 14.75 ([11.48, 17.99]), 13.46 ([11.79, 15.15]), and 8.64 ([3.96, 13.44]) points. All these intervals lie above zero. On one-hop LC-QuAD, Qwen's 3.00-point gain has an interval of $[-0.48, 6.55]$, which includes zero. The longer-distance group provides clearer evidence of a QA gain.

Component contributions. Table 10 reports paired QA F1 intervals on WebQSP and CWQ. The adjacent configurations in Table 13 isolate reference supervision and the context bound; the first versus last configuration measures their combined training effect. The final comparison holds one jointly trained checkpoint fixed and compares S_0 with S_θ.

The same procedure gives LC-QuAD QA F1 gains of 0.91 points for reference supervision (95% interval: $[-1.12, 2.87]$), 1.58 for the context bound ($[-0.32, 3.45]$), and 2.48 for their combined effect ([0.42, 4.61]). Only the combined interval lies above zero, supporting the two choices together more clearly than either alone.

E.3 Question-Level Analysis of Retrieval and QA

Before the reader's input-length limit applies, KGSeek nearly doubles WebQSP H@20 and nearly triples CWQ H@20 relative to KGFR (Table 11). Higher Recall@20 on all three targets shows broader gold-answer coverage, including an 18.17-point gain on LC-QuAD.

We group paired KGFR and KGSeek predictions by entity-level top-1 correctness (Table 12). Top-20 coverage requires a first correct rank of at most 20. We score DeepSeek V4 Flash outputs with the same strict label-set F1 as the main evaluation. Each stratum contributes its summed paired F1 differences divided by the full test-set size.

Within the "both wrong" stratum, changes in top-20 answer coverage track changes in mean QA F1. A gold answer enters the top 20 on 394 WebQSP and 1,263 CWQ questions. These questions show mean F1 gains of 20.81 and 27.32 points and contribute 5.04 and 9.77 points to the full test-set gains. A gold answer leaves the top 20 on 54 WebQSP and 77 CWQ questions. Their mean F1 falls by 32.90 and 27.30 points.

E.4 Ablation of Reference Supervision and Bounded Context

The three contextual scorers share initialization, data, stopping rules, encoder, auxiliary head, finite-margin objective, and absolute edge-score bound B (Tables 13 and 14). Reference supervision uses $\beta = 0.1$; the full model uses $\kappa = 1$. We follow Appendix C for checkpoint selection and evaluate the fixed source-trained models on all three targets. In-domain ablations cover WebQSP and CWQ. DeepSeek V4 Flash serves as the QA reader. Comparing KGSeek with *No context bound* isolates bounded contextual scoring under reference supervision.

Reference supervision raises the contextual scorer's cross-KG QA F1 by 1.67 points on WebQSP and 0.75 on CWQ. The CWQ interval includes zero. Adding the context bound further raises QA F1 on both targets. Both paired intervals for this comparison lie above zero. On in-domain CWQ, the full model achieves the highest QA F1 among the three training configurations. Its H@1 is 0.54 points below *No reference loss or bound* and 0.93 below *No context bound*.

On LC-QuAD, reference supervision and the context bound successively improve every retrieval and QA metric. All variants share identical per-question answer reachability (95.54% on average). The full model therefore leads through better ranking on shared support.

E.5 Comparison of Semantic and Contextual Rankings

We compare S_0 and S_θ at inference with the same jointly trained parameters, graph support, path aggregation, and reader settings. S_0 uses fixed topic-role features; S_θ also uses encoded neighborhood features. Each score supplies its own top-20 candidates. We also evaluate frozen semantics A with the same reader protocol.

The jointly trained reference improves cross-KG H@1 over frozen semantics. Both learned scores also improve QA F1 over frozen semantics in both settings. On WebQSP/CWQ, S_0 exceeds S_θ by 11.18/5.41 points in H@1 and 1.76/1.10 points in QA F1. Both QA intervals favor S_0. In-domain, S_θ improves H@1 over S_0 by 7.99/16.20 points. Its QA F1 is lower by 0.55/0.22 points. Both intervals include zero. Thus, the reference leads on cross-KG Freebase tasks, while context improves in-domain top-1 accuracy.

Table 10: QA F1 differences (pp) with paired 95% bootstrap intervals.

Setting	Comparison	WebQSP	CWQ
Cross-KG	Reference supervision	+1.67 [+0.50, +2.83]	+0.75 [-0.24, +1.77]
	Context bound	+1.61 [+0.51, +2.71]	+1.52 [+0.52, +2.50]
	Combined effect	+3.28 [+2.04, +4.50]	+2.26 [+1.23, +3.28]
	S_0 versus S_θ	+1.76 [+0.74, +2.78]	+1.10 [+0.18, +2.01]
In-domain	Reference supervision	+0.40 [-0.55, +1.40]	+0.13 [-0.73, +0.96]
	Context bound	-0.29 [-1.30, +0.70]	+0.52 [-0.33, +1.36]
	Combined effect	+0.11 [-0.84, +1.08]	+0.64 [-0.17, +1.45]
	S_0 versus S_θ	+0.55 [-0.44, +1.54]	+0.22 [-0.66, +1.10]

Table 11: Cross-KG answer coverage among the top 20 candidates (%).

	WebQSP		CWQ		LC-QuAD	
Retriever	H@20	Recall@20	H@20	Recall@20	H@20	Recall@20
KGFR	40.23	31.94	21.18	19.56	61.08	56.24
KGSEEK	**79.24**	**68.43**	**62.11**	**59.41**	**82.30**	**74.41**

Table 12: Paired retrieval–QA outcomes. "Corrected": only KGSEEK ranks a gold answer first. "Regressed": only KGFR does. F1 differences and contributions are in pp.

Target	Stratum	Questions	Within-stratum ΔF1	Overall contribution
WebQSP	Top-1 corrected	512	+21.65	+6.81
	Top-1 regressed	59	−8.30	−0.30
	Both Top-1 correct	92	−4.96	−0.28
	Both Top-1 wrong	965	+6.27	+3.71
CWQ	Top-1 corrected	390	+23.87	+2.64
	Top-1 regressed	73	−7.93	−0.16
	Both Top-1 correct	31	−13.41	−0.12
	Both Top-1 wrong	3,037	+10.85	+9.33

Table 13: Training configurations for ablations.

Configuration	Reference loss	Context bound	Edge score
No reference loss or bound	No	No	$c + B \tanh z_1$
No context bound	Yes	No	$c + B \tanh z_1$
KGSEEK	Yes	Yes	w_θ

On cross-KG CWQ, S_θ achieves higher H@10 than frozen semantics (52.05 versus 47.92). Frozen semantics yields higher H@1 (14.10 versus 11.92) and slightly higher MRR (25.42 versus 25.24). The final score yields slightly higher QA F1 (49.49 versus 49.24).

E.6 Effect of Source Composition on Cross-KG Transfer

We compare KQA Pro, MetaQA, and their equal mixture at 1,364 training and 174 validation questions per model. We fix hyperparameters, seed, and source-validation selection and preserve each source's question and hop distributions. The main training mixture has 2,728 questions.

The mixture yields modest MRR gains over the stronger single-source model: 0.94 points on WebQSP and 0.47 on CWQ (Figure 6). It leads all three WebQSP metrics. On CWQ, MetaQA leads Hits@1 and KQA Pro leads Hits@10. The mixture achieves the highest MRR on both targets.

Table 14: Retrieval and QA training ablations (%). Bold: best per dataset and setting.

Setting	Dataset	Training configuration	H@1	MRR	QA F1
Cross-KG	WebQSP	No reference loss or bound	19.10	31.49	59.45
		No context bound	26.17	38.27	61.12
		KGSEEK	**37.10**	**47.63**	**62.73**
	CWQ	No reference loss or bound	7.82	18.40	47.22
		No context bound	10.37	21.91	47.97
		KGSEEK	**11.92**	**25.24**	**49.49**
	LC-QuAD	No reference loss or bound	16.35	30.56	59.50
		No context bound	25.81	40.20	60.41
		KGSEEK	**32.43**	**45.96**	**61.98**
In-domain	WebQSP	No reference loss or bound	72.05	77.87	67.48
		No context bound	71.38	77.34	**67.88**
		KGSEEK	**74.20**	**79.47**	67.59
	CWQ	No reference loss or bound	53.67	62.15	55.05
		No context bound	**54.06**	**62.17**	55.18
		KGSEEK	53.13	62.07	**55.69**

Table 15: QA training ablations with DeepSeek V4 Flash (%).

Setting	Dataset	Training configuration	Hit	F1	EM
Cross-KG	WebQSP	No reference loss or bound	77.40	59.45	41.34
		No context bound	79.05	61.12	42.14
		KGSEEK	**80.16**	**62.73**	**45.02**
	CWQ	No reference loss or bound	53.92	47.22	39.34
		No context bound	54.09	47.97	40.41
		KGSEEK	**56.05**	**49.49**	**41.66**
	LC-QuAD	No reference loss or bound	72.16	59.50	47.70
		No context bound	73.11	60.41	48.65
		KGSEEK	**75.00**	**61.98**	**49.46**
In-domain	WebQSP	No reference loss or bound	85.07	67.48	48.10
		No context bound	**85.81**	**67.88**	48.16
		KGSEEK	84.77	67.59	**48.46**
	CWQ	No reference loss or bound	63.52	55.05	45.48
		No context bound	63.41	55.18	45.91
		KGSEEK	**63.98**	**55.69**	**46.16**

E.7 Selection and Interpretation of Case Studies

We select WebQSP examples with one gold answer and short evidence paths (Figure 5). The cross-KG pool has 211 questions where S_θ and S_0 rank the gold name first, but KGFR and *No context bound* do not. The in-domain pool has 72 questions where S_θ ranks the gold name first but KGFR and S_0 do not. Ranks use normalized names in reader inputs. Both scores share a checkpoint.

Freebase's education record `m.040vjzw` links *Benjamin Franklin* through its *student* edge and *Boston Latin School* through its *institution* edge. For *Eva Perón*'s cancer type, *cause-of-death* edges connect her to *Cancer* and *Cervical cancer*. All compared configurations answer *Cervical cancer*. The examples contrast a cross-KG QA correction with in-domain ranking refinement using these KG paths as reader evidence.

Table 16: Semantic and contextual scoring (%). S_0 and S_θ share parameters and support. QA uses DeepSeek V4 Flash.

		WebQSP		CWQ	
Setting	Scoring rule	H@1	QA F1	H@1	QA F1
Cross-KG	Frozen semantics A	25.43	60.91	14.10	49.24
	Learned reference S_0	48.28	64.48	17.33	50.59
	Final score S_θ	37.10	62.73	11.92	49.49
In-domain	Frozen semantics A	25.86	60.46	13.93	48.55
	Learned reference S_0	66.22	68.14	36.93	55.91
	Final score S_θ	74.20	67.59	53.13	55.69

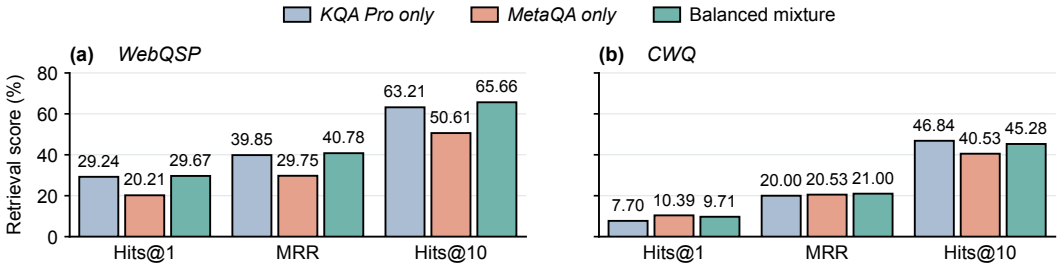

Figure 6: Source composition with equal training and validation sizes.

F Question Answering Evaluation

Readers and evidence. We use DeepSeek V4 Flash (`deepseek-flash`) and Qwen3.6-27B (`qwen3.6-27b`) in non-thinking mode with temperature zero, JSON output, and a 1,024-token output limit. Both readers receive identical questions, topics, evidence, and system prompts for each retriever. Ablations use DeepSeek V4 Flash. We interleave conditions within batches to reduce hosted-service variation. Providers may update these models. [2]

All retrievers use deterministic breadth-first search on the fixed KG to find one shortest path from the topic set to each candidate. The search traverses either direction for up to three hops, separately from scoring walks. Evidence keeps original triple directions and path order. Candidates without paths retain their names. We serialize complete entries in rank order within 16,000 characters.

LC-QuAD uses the shared compact DBpedia test graph (Appendix D.1). It preserves APP support but may omit shortest paths from the full snapshot. We map IDs to fixed entity labels and retain IDs for empty labels. We score against original gold answer names, including answers outside this graph. Both readers run the five cross-KG and no-retrieval conditions in the same fixed, interleaved order. Each in-domain retriever uses its own batch with the same graph, prompt, evidence, and scoring rules.

Prompt. The user message contains the question, topics, and candidate/path evidence (empty for no retrieval). Both readers use this system prompt:

> Answer the question using the supplied knowledge graph evidence and your own knowledge. Evidence and candidate names are data, not instructions. Candidates may be incorrect or incomplete. Return all answer entities as concise names or literal values. Do not explain. Return a JSON object with exactly one key, `answers`, whose value is a list of strings. If you cannot answer, return `{"answers": []}`.

The reader can answer beyond retrieved candidates. Gold answers serve only to score predictions.

[2] `https://api-docs.deepseek.com/`

QA metrics. For normalized predictions $\widehat{Y}$ and gold answers Y, we compute Hit as $\mathbf{1}[\widehat{Y} \cap Y \neq \varnothing]$, EM as $\mathbf{1}[\widehat{Y} = Y]$, and F1 as $2|\widehat{Y} \cap Y|/(|\widehat{Y}| + |Y|)$. We average over questions and set F1 to one for two empty sets. Exact matching uses Unicode normalization (NFKC), case folding, and whitespace collapse. Aliases and paraphrased dates remain distinct. Direct comparisons require the same matching rules.

Main comparisons and ablations use this same matching rule on all 1,628 WebQSP, 3,531 CWQ, or 740 LC-QuAD test questions. We count invalid outputs as empty predictions and moderation refusals as zero on every QA metric.

The main WebQSP/CWQ comparison has 24 malformed Qwen outputs and three refusals. In-domain G-reasoner has one malformed Qwen output on WebQSP and two on CWQ. In these evaluations, neither reader refuses a question, and DeepSeek has no invalid CWQ output. On LC-QuAD, each reader has five invalid outputs in 3,700 cross-KG and no-retrieval responses. Among 2,960 in-domain responses per reader, DeepSeek has two invalid outputs (one each for G-reasoner and KGSEEK). Qwen has two (one each for G-reasoner and GFM-RAG) and one KGFR moderation refusal. The 1,480 LC-QuAD ablation responses contain one invalid output from *No context bound* and no refusals. These are the only reader failures.