EAI Endorsed Transactions
on Smart Cities
Research Article EAI.EU

CERA-OSR: Counterfactual Evidence Routing for Auditable One-Shot Subgraph Reasoning in Smart-City Knowledge Graph Completion

Van Ba Nguyen[1], Minh Y. Nguyen[1], Ha-Phuong Nguyen[2], Thuy-A Nguyen[1*]

[1]Faculty of Information Technology, Ho Chi Minh City University of Foreign Languages-Information Technology, Ho Chi Minh City, Vietnam

[2]School of Engineering and Technology, Hue University

Abstract

Knowledge graph completion supports data integration, while compact subgraph retrieval can overlook relevant evidence. CERA-OSR integrates train-only multi-view evidence pooling, relation-dependency priors, query-conditioned soft routing, uncertainty-triggered recovery and counterfactual evidence training within a one-shot subgraph reasoning framework. On WN18RR, FB15k-237, NELL-995 and YAGO3-10, CERA-OSR achieves MRR values of 0.626, 0.468, 0.603 and 0.676, respectively. The unweighted mean MRR is 0.5933, compared with 0.5738 for MoKGR, an absolute improvement of 0.0195. Ablation and graph-perturbation analyses examine component contributions and prediction sensitivity to selected evidence. Efficiency measurements reveal a trade-off between ranking performance and the cost of conditional recovery. The framework is motivated by smart-city knowledge graphs and evaluated on general knowledge graph benchmarks.

Received on 23 May 2026; accepted on 08 November 2026; published on 10 November 2026

Keywords: knowledge graph reasoning, smart cities, one-shot subgraph reasoning, evidence routing, counterfactual learning

Copyright © 2026 Van Ba Nguyen *et al.*, licensed to EAI. This is an open access article distributed under the terms of the CC BY-NC-SA 4.0, which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi:10.4108/eetsc.13147

1. Introduction

Smart-city knowledge graphs connect entities such as roads, public agencies, sensors, services and incidents. Completing missing relations could support semantic search and data integration, but an operational user may also need to inspect the graph evidence behind a prediction. This motivates a KG reasoning framework that exposes selected evidence and permits uncertain selections to be reconsidered.

For example, predicting which agency manages a road segment involves ownership and administrative relations, whereas proximity to a highly connected transport hub may provide less relevant evidence. This scenario motivates relation-aware selection and reconsideration of evidence under uncertain rankings.

Embedding models such as ConvE, RotatE and QuatE score triples through learned representations [1–3]. Path and GNN methods exploit relational structure [4–10]. One-shot-subgraph reasoning separates query-dependent retrieval from prediction on the extracted region [11], while MoKGR personalizes path length and pruning through expert mixtures [12]. These approaches motivate the compact reasoning backbone considered here.

A remaining challenge is to reconsider low-weight evidence after an uncertain first pass and assess how selected edges influence a prediction. Continuous gates expose evidence weights, while perturbation tests evaluate the sensitivity of model predictions to changes in that evidence.

CERA-OSR combines a relation-aware candidate pool with continuous edge gates and a bounded recovery pass that revisits low-gated evidence when the

*Corresponding author. Email: antt@huflit.edu.vn

first ranking is uncertain. Counterfactual training contrasts the original evidence with corrupted alternatives, making prediction sensitivity to selected evidence an explicit learning target. The methodological contribution is the integration of retrieval, weighting, recovery and perturbation testing within a one-shot subgraph reasoning framework.

Section 2 reviews related work. Section 3 presents the framework, Section 4 describes the experimental setup, and Section 5 reports ranking performance, ablation, evidence sensitivity and computational cost. Section 6 discusses application scenarios, limitations and future work, followed by the conclusion in Section 7.

The methodological contributions are as follows:

1. A train-only candidate pool combines proximity, relation and path signals to provide the search space for query-conditioned routing.
2. Continuous edge gates modulate message passing, with a soft budget controlling their aggregate weight.
3. A bounded recovery pass revisits low-gated candidates when the initial ranking is uncertain, without retrieving held-out graph edges.
4. Counterfactual training contrasts original evidence with relation, path and distractor perturbations; component removal and perturbation tests examine its effect on ranking and prediction sensitivity.
5. The evaluation measures ranking accuracy, the effects of component removal, evidence sensitivity, recovery transitions and computational cost.

2. Related Work

2.1. Knowledge Graph Reasoning for Smart Cities

Urban KGs integrate entities and relations associated with locations, infrastructure, services, mobility and events. UUKG combines urban data from multiple sources for knowledge-enhanced spatiotemporal prediction [13]. In this setting, inspecting the entities and relations that support a prediction can assist the interpretation of model outputs.

Urban graphs often contain unevenly observed relation types and heterogeneous data sources. Combining structural proximity with relation compatibility can help identify candidate paths relevant to a query.

For a given query relation, a locally dense region may contain both useful and unrelated paths. Relation-conditioned routing weights these candidates, while deletion and counterfactual tests assess their influence on the predicted score.

Continuous edge gates, recovered candidates and changes in predicted scores provide complementary records of the evidence-selection process. These records support inspection of the model's reasoning at the edge and path levels.

2.2. Subgraph and Adaptive Path Reasoning

Embedding-based KG completion models score triples using entity and relation representations [1–3]. Their scores do not directly identify supporting graph paths. Rule, reinforcement-learning and differentiable path methods, including MINERVA, DRUM and AnyBURL, represent relational patterns or traversal paths explicitly [4, 5, 14]. GNN-based reasoners such as CompGCN, NBFNet and RED-GNN propagate query information through relational neighborhoods [6–8].

One-shot-subgraph reasoning extracts a query-dependent subgraph before prediction, using PPR as a retrieval heuristic [11]. MoKGR uses mixtures of length and pruning experts for personalized exploration [12]. CERA-OSR extends compact retrieval with continuous evidence weights, bounded recovery of low-gated candidates and training against corrupted evidence.

CERA-OSR combines training-derived structural, relation and path signals during retrieval. Continuous edge gates then modulate message passing, and a depth mixture combines expert outputs. An uncertain first-pass ranking triggers reconsideration of low-gated candidates within the initial pool. Counterfactual training contrasts predictions obtained from original and corrupted evidence.

2.3. Explanation Evaluation and Urban KG Construction

PaGE-Link provides path explanations for heterogeneous link prediction [15]. GraphFramEx distinguishes necessary and sufficient prediction fidelity in node classification [16]. These studies motivate evaluating evidence influence separately from ranking accuracy, although their task settings differ from knowledge graph completion.

UrbanKGent addresses urban KG construction using an LLM agent framework [17], while UUKG provides graph data for knowledge-enhanced urban spatiotemporal tasks [13]. These works characterize data and task requirements for urban applications. NIST AI RMF 1.0 [18] places trustworthy AI evaluation in its application context, supporting consideration of evidence quality and operational interpretation alongside predictive performance.

3. Methodology

3.1. Problem Formulation

Let a knowledge graph be denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$, where $\mathcal{V}$ is the entity set, $\mathcal{R}$ is the relation set and $\mathcal{E}$ is the

Table 1. Taxonomy of knowledge graph reasoning methods and positioning of CERA-OSR

Group	Representative methods	Main mechanism
Embedding KGC	ConvE, QuatE, RotatE	Entity and relation embedding scoring
Path, RL and rule	MINERVA, DRUM, AnyBURL	Rule or path search
Full GNN reasoning	NBFNet, RED-GNN, CompGCN	Message passing over KG neighborhoods
Adaptive pruning GNN	AdaProp, A*Net	Learned or prioritized propagation
One-shot subgraph	one-shot-subgraph	PPR query subgraph plus predictor
Mixture experts	MoKGR	Adaptive length and pruning experts
CERA-OSR	This work	Counterfactual, recoverable, relation-guided evidence routing

set of observed triples. A triple is (h, r, t), with $h, t \in \mathcal{V}$ and $r \in \mathcal{R}$. We focus on tail entity prediction. Given a query $q = (h, r_q, ?)$, the model ranks all candidate tails $t \in \mathcal{V}$ so that valid triples (h, r_q, t) are placed near the top. The scoring function is

$$s_q(t) = f_\Theta(h, r_q, t, \mathcal{G}_{\text{train}}), \tag{1}$$

where Θ denotes model parameters and $\mathcal{G}_{\text{train}}$ is the training graph only. All evidence pools, path patterns, relation dependencies, recovery candidates and counterfactual graphs are constructed from $\mathcal{G}_{\text{train}}$, which prevents leakage from validation or test triples into evidence generation.

Under filtered ranking, other known correct answers are excluded using the train, validation and test fact lists. Held-out facts are used for evaluation filtering; evidence construction and counterfactual training use only the training graph. The primary metrics are mean reciprocal rank (MRR), Hits@1 and Hits@10.

The evaluation is transductive: candidate entities belong to a fixed entity universe, while validation and test triples are held out.

Validation metrics guide model selection. Evidence features, relation dependencies, recovery candidates and cached training-query signals are derived from training data.

CERA-OSR constructs an initial pool of candidate edges and paths from the training graph and updates their contribution to reasoning through query-conditioned weights. Each edge is represented in the context of the current query, allowing its relevance to vary across relations. The candidate pool remains fixed during inference, while soft routing and recovery modify the influence of its edges on the prediction.

Gate inspection, edge deletion and recovery transitions characterize evidence weighting, prediction sensitivity and changes in ranking, respectively.

3.2. Train-Only Multi-View Evidence Pool

For each query q, CERA-OSR constructs a candidate evidence graph

$$\mathcal{G}_q^0 = (\mathcal{V}_q^0, \mathcal{E}_q^0), \tag{2}$$

where $\mathcal{V}_q^0$ and $\mathcal{E}_q^0$ are generated from $\mathcal{G}_{\text{train}}$. Instead of relying on a single retrieval signal such as PPR, CERA-OSR combines multiple complementary views: PPR proximity from the head entity, bounded breadth-first-search depth, a relation prior from the RDG, a path prior from training graph patterns, a degree penalty to reduce hub bias and cached uncertainty cues from the scorer during training.

For an edge e, the pool score is

$$\begin{aligned} s_{\text{pool}}(e \mid q) = \alpha_1 s_{\text{ppr}} + \alpha_2 s_{\text{rdg}} + \alpha_3 s_{\text{path}} \\ + \alpha_4 s_{\text{dist}} - \alpha_5 s_{\text{deg}} + \alpha_6 s_{\text{unc}}, \end{aligned} \tag{3}$$

In Eq. (3), s_{ppr} is head-personalized PageRank proximity; s_{rdg} is relation attention as defined in Eq. (5); s_{path} summarizes training-path pattern frequency or confidence; s_{dist} is proximity derived from bounded BFS distance; s_{deg} penalizes normalized node degree; and s_{unc} summarizes cached first-pass uncertainty from training queries. The α coefficients in this equation weight retrieval signals and are distinct from the depth-mixture weights in Eq. (11).

The retrieval views serve complementary roles: PPR favors local proximity, bounded BFS limits expansion, relation and path priors emphasize query-compatible patterns, and the degree penalty reduces hub dominance. The coefficients in Eq. (3) control the relative contribution of these signals.

The multi-view pool separates retrieval from subsequent evidence weighting. A broad candidate region preserves potential support paths, while the learned router controls their contribution to message passing. Recovery revisits candidates in this training-derived pool.

Cached uncertainty is a training-time signal derived from earlier scorer predictions. It identifies graph regions associated with unstable first-pass rankings and complements the structural retrieval views.

3.3. Relation-Dependency Graph and Edge Tokenizer

CERA-OSR builds a Relation-Dependency Graph

$$\mathcal{G}_R = (\mathcal{R}, \mathcal{E}_R), \tag{4}$$

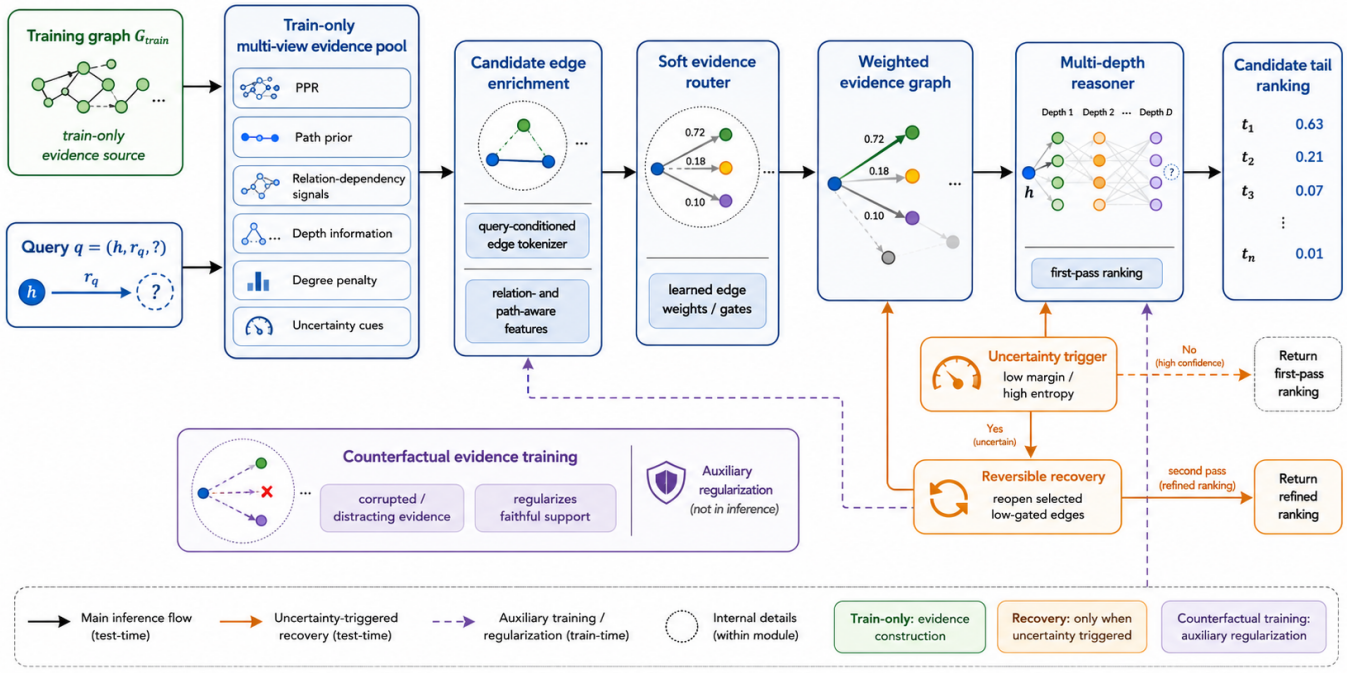


Figure 1. Architecture of CERA-OSR, comprising a train-only multi-view evidence pool, relation-dependency encoding, soft evidence routing, multi-depth reasoning, uncertainty-triggered recovery and counterfactual evidence training

Each RDG node represents a relation type. Directed dependencies summarize relation transitions, local co-occurrence and inverse patterns observed in training paths, supplying a relation-compatibility prior for the current query.

Given query relation r_q, the RDG encoder computes attention $a_{q,r}$ between r_q and each relation r. For edge $e = (u, r, v)$, the relation prior is

$$s_{\mathrm{rdg}}(e \mid q) = a_{q,r}. \tag{5}$$

Each edge is tokenized into $x_e(q)$ by concatenating embeddings of u, v, r, r_q and the structural signals described above. The same edge can therefore become evidence or noise depending on the query context.

The RDG provides a learned compatibility prior rather than a fixed symbolic rule base. Query-conditioned attention makes the contribution of an edge relation dependent on the target relation.

3.4. Budget-Constrained Soft Evidence Router

Instead of hard pruning, CERA-OSR learns a soft gate for every edge:

$$g_e(q) = \sigma\left(\mathrm{MLP}_g(x_e(q))\right). \tag{6}$$

The gate modulates message passing:

$$m^{(\ell)}_{u\to v} = g_e(q) \cdot \phi_\ell\left(h^{(\ell)}_u, z_r, z_{r_q}\right), \tag{7}$$

Here z_r and z_{r_q} are relation embeddings and ϕ is the message function at the specified layer. Equation (8) defines the total gate mass over the finite initial edge pool:

$$B_q = \sum_{e\in\mathcal{E}^0_q} g_e(q) \tag{8}$$

and the budget loss is

$$\mathcal{L}_{\mathrm{budget}} = \left(\frac{B_q}{|\mathcal{E}^0_q|} - \rho\right)^2, \tag{9}$$

In Eq. (9), ρ is the target mean gate value and the denominator is the initial pool size. Since each gate lies between zero and one, B_q lies between zero and the pool size. The budget measures total evidence weight, whereas physical edge count and execution cost are evaluated separately.

The squared budget penalty regularizes the mean gate value toward the target budget. Low-weight candidates remain in the initial pool and are eligible for recovery.

3.5. Multi-Depth Reasoner and First-Pass Scoring

Each depth expert produces query-conditioned node representations in a common hidden dimension over the same candidate-entity indexing:

$$H^{(k)}_q = \mathrm{Reasoner}^{(k)}(\mathcal{G}^g_q, q), \tag{10}$$

The first-pass graph uses soft edge weights. A depth router maps the query relation and pooled evidence

statistics to logits; a softmax over the finite depth set gives nonnegative mixture weights summing to one. The mixture is

$$H_q = \sum_{k \in \mathcal{K}} \alpha_k(q) H_q^{(k)} \tag{11}$$

Candidate tails are scored as

$$s_q^{(1)}(t) = \mathrm{MLP}_s\left([h_t, H_q(t), z_{r_q}, z_h]\right). \tag{12}$$

The first-pass ranking supplies the two highest-scoring candidates, their score margin and a normalized candidate-score distribution for entropy. Recovery uses these model outputs without access to the gold test answer.

The depth experts share node indexing and hidden dimensions, allowing their representations to be combined in Eq. (11). In Eq. (12), the candidate-tail representation is selected from the combined node representations before scoring.

3.6. Reversible Evidence Recovery

Let t_1 and t_2 denote the first and second candidates under descending first-pass scores. Recovery is triggered when the top-two score margin is below the margin threshold or Shannon ranking entropy exceeds the entropy threshold. The margin condition is

$$\Delta_q = s_q^{(1)}(t_1) - s_q^{(1)}(t_2) < \tau_m, \tag{13}$$

Ranking entropy is Shannon entropy computed from a softmax-normalized candidate-score distribution using natural logarithms. It is expressed in nats, with a maximum equal to the logarithm of the candidate-set size. The low-gated candidate pool is

$$\mathcal{E}_q^- = \{e \in \mathcal{E}_q^0 : g_e(q) < \tau_g\}. \tag{14}$$

A recovery MLP scores candidates using the edge token, original gate, relation prior, path prior, proximity to first-pass top candidates and uncertainty features. Top-B_{rec} denotes at most B_{rec} candidates with highest recovery scores, capped by the available low-gated pool. Reopened evidence is restricted to the original training-derived pool:

$$\mathcal{E}_q^{\mathrm{rec}} = \mathrm{Top}_{B_{\mathrm{rec}},\, e \in \mathcal{E}_q^-} r_e(q). \tag{15}$$

Equation (15) specifies the recovered support within the original candidate pool. Selected low-gated edges are reactivated for the second pass, increasing their contribution to message passing. Recovery does not retrieve additional edges outside the initial pool. The second-pass score is combined as

$$s_q(t) = \lambda_q s_q^{(2)}(t) + (1 - \lambda_q) s_q^{(1)}(t). \tag{16}$$

The mixture coefficient λ_q in Eq. (16) lies in $[0, 1]$ and controls the contribution of the second-pass score. Queries that do not trigger recovery retain their first-pass scores.

The margin, entropy and low-gate thresholds are 0.05, 1.20 and 0.30, respectively, and the recovery budget ratio is 0.03.

3.7. Counterfactual Evidence Training

Counterfactual training contrasts original evidence with controlled graph perturbations, including support-edge removal, relation swapping, path corruption and distractor insertion. These transformations change the evidence available to the reasoner and encourage discrimination between supportive and misleading graph inputs.

Counterfactual graphs are constructed from training evidence, and corrupted triples are filtered against known training facts to reduce false negatives. Path corruption modifies the relational composition of the evidence paths. The original and perturbed evidence graphs define the counterfactual loss:

$$\mathcal{L}_{\mathrm{cf}} = \max\left(0, \gamma - s_q(t^+ \mid \mathcal{G}_q^+) + s_q(t^+ \mid \mathcal{G}_q^-)\right). \tag{17}$$

The distractor loss penalizes high scores produced by distracting evidence:

$$\begin{aligned} \mathcal{L}_{\mathrm{dist}} = \max\big(0, \gamma_d - s_q(t^+ \mid \mathcal{G}_q^+) \\ + s_q(t^- \mid \mathcal{G}_q^{\mathrm{distractor}})\big). \end{aligned} \tag{18}$$

The complete objective is

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{\mathrm{rank}} + \lambda_b \mathcal{L}_{\mathrm{budget}} + \lambda_{\mathrm{cf}} \mathcal{L}_{\mathrm{cf}} + \lambda_d \mathcal{L}_{\mathrm{dist}} \\ + \lambda_g \mathcal{L}_{\mathrm{gate}} + \lambda_r \mathcal{L}_{\mathrm{recovery}} \\ + \lambda_s \mathcal{L}_{\mathrm{stability}} + \lambda_k \mathcal{L}_{\mathrm{depth}}. \end{aligned} \tag{19}$$

In Eq. (19), the ranking term supervises link prediction. The budget loss regulates evidence weight, and the hinge losses in Eqs. (17)–(18) contrast supportive, corrupted and distracting evidence. The remaining auxiliary terms address gate supervision, recovery supervision, score stability and depth regularization. Their coefficients balance the contributions to the joint objective.

4. Experimental Setup

4.1. Datasets and Metrics

The evaluation covers Family, UMLS, WN18RR, FB15k-237, NELL-995 and YAGO3-10. Dataset provenance includes WN18RR from the ConvE study [1], FB15k-237 from the Freebase benchmark of Toutanova and Chen [19], and YAGO3 underlying YAGO3-10

Table 2. Dataset statistics for the study partitions. Train denotes training queries; retrieval facts are maintained separately

Dataset	Entities	Relations	Train	Valid	Test
Family	3,007	12	5,868	2,038	2,835
UMLS	135	46	1,321	569	633
WN18RR	40,943	11	21,705	3,034	3,134
FB15k-237	14,541	237	68,028	17,535	20,466
NELL-995	74,536	200	37,420	543	2,818
YAGO3-10	123,188	37	269,760	5,000	5,000

[20]. WN18RR, FB15k-237, NELL-995 and YAGO3-10 constitute the four core benchmarks. Table 2 summarizes the dataset partitions used in the study.

For each query, the model ranks candidate tail entities and filters other known correct answers when computing the evaluation metrics. Table 3 reports MRR and Hits metrics; Tables 5–8 report perturbation, recovery, computational cost and sparse-query results. The core average is the unweighted mean over WN18RR, FB15k-237, NELL-995 and YAGO3-10.

4.2. Baselines and Protocol

The baselines include embedding methods (ConvE, QuatE and RotatE), path and rule methods (MINERVA, DRUM and AnyBURL), GNN reasoners (CompGCN, NBFNet and RED-GNN) and adaptive propagation methods (AdaProp and A*Net). One-shot-subgraph reasoning provides the compact retrieval-and-prediction backbone. MoKGR provides a direct comparison with adaptive reasoning depth and pruning through expert mixtures.

Table 3 uses literature baseline values from the transductive comparison in MoKGR [12] and CERA-OSR results from this study. MoKGR and one-shot-subgraph reasoning provide the closest methodological comparisons because they also use query-dependent compact reasoning.

4.3. Implementation and Reproducibility

Evidence-pool construction, RDG generation, path priors, recovery candidates and counterfactual graphs use the training graph. Validation metrics guide hyperparameter selection, while test facts are used only for evaluation filtering.

The configuration uses AdamW with learning rate 5×10^{-4}, weight decay 10^{-5}, hidden and relation dimensions of 256, dropout 0.10, 64 negatives per query, expert depths $\{2, 4, 6, 8\}$ and target mean gate value 0.50. Experiments use five random seeds $\{1, 2, 3, 4, 5\}$ on an NVIDIA L40 GPU with approximately 45 GB usable memory. Recovery settings are margin threshold 0.05, entropy threshold 1.20, low-gate threshold 0.30 and budget ratio 0.03.

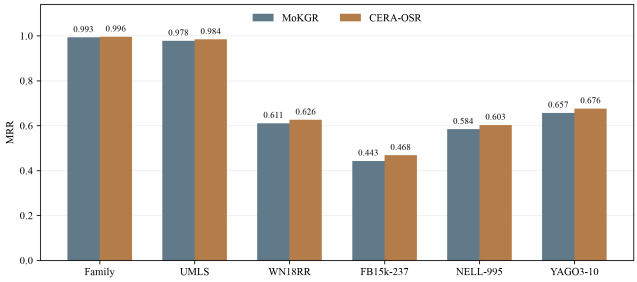


Figure 2. MRR comparison between MoKGR [12] and CERA-OSR on six benchmarks

All component-removal variants reuse the evidence-pool generation procedure and filtered-ranking protocol, keeping candidate generation consistent with the full model.

5. Results

5.1. Link Prediction Performance

Table 3 reports CERA-OSR MRR values of 0.626, 0.468, 0.603 and 0.676 on WN18RR, FB15k-237, NELL-995 and YAGO3-10, respectively. The corresponding MoKGR values are 0.611, 0.443, 0.584 and 0.657, giving absolute improvements of 0.015, 0.025, 0.019 and 0.019. The unweighted core averages are 0.5933 for CERA-OSR and 0.5738 for MoKGR, with an absolute difference of 0.0195.

CERA-OSR achieves the highest MRR, Hits@1 and Hits@10 among the methods compared in Table 3 on all six datasets. The largest core MRR improvement over MoKGR occurs on FB15k-237. Table 4 examines the contribution of individual components.

Figure 2 compares the MRR of MoKGR and CERA-OSR across the six benchmarks.

5.2. Ablation Study

Table 4 reports single-component removals on the four core datasets. The full-model average is 0.5933. Removing depth selection, counterfactual training, recovery, the soft router and the RDG prior reduces the unrounded core average by 0.00625, 0.01575, 0.03025, 0.07150 and 0.08700, respectively.

Removing depth selection produces the smallest mean MRR decrease among the evaluated removals, while removing the RDG prior produces the largest. The decreases associated with soft routing and recovery also indicate their contribution within the full architecture. Each ablation measures the effect of removing one component while retaining the others; these comparisons do not isolate joint effects among components.

Table 3. Filtered link prediction results. Entries are MRR/Hits@1/Hits@10, with MRR in decimals and Hits in percentages. Baseline values are from [12]; CERA-OSR entries are five-seed means

Type	Model	Family	UMLS	WN18RR	FB15k-237	NELL-995	YAGO3-10
Non-GNN	ConvE	0.912/83.7/98.2	0.937/92.2/96.7	0.427/39.2/49.8	0.325/23.7/50.1	0.511/44.6/61.9	0.520/45.0/66.0
Non-GNN	QuatE	0.941/89.6/99.1	0.944/90.5/99.3	0.480/44.0/55.1	0.350/25.6/53.8	0.533/46.6/64.3	0.379/30.1/53.4
Non-GNN	RotatE	0.921/86.6/98.8	0.925/86.3/99.3	0.477/42.8/57.1	0.337/24.1/53.3	0.508/44.8/60.8	0.495/40.2/67.0
Non-GNN	MINERVA	0.885/82.5/96.1	0.825/72.8/96.8	0.448/41.3/51.3	0.293/21.7/45.6	0.513/41.3/63.7	–
Non-GNN	DRUM	0.934/88.1/99.6	0.813/67.4/97.6	0.486/42.5/58.6	0.343/25.5/51.6	0.532/46.0/66.2	0.531/45.3/67.6
Non-GNN	AnyBURL	0.861/87.4/89.2	0.828/68.9/95.8	0.471/44.1/55.2	0.301/20.9/47.3	0.398/27.6/45.4	0.542/47.7/67.3
GNN	CompGCN	0.933/88.3/99.1	0.927/86.7/99.4	0.479/44.3/54.6	0.355/26.4/53.5	0.463/38.3/59.6	0.421/39.2/57.7
GNN	NBFNet	0.989/98.8/98.9	0.948/92.0/99.5	0.551/49.7/66.6	0.415/32.1/59.9	0.525/45.1/63.9	0.550/47.9/68.6
GNN	RED-GNN	0.992/98.8/99.7	0.964/94.6/99.0	0.533/48.5/62.4	0.374/28.3/55.8	0.543/47.6/65.1	0.559/48.3/68.9
GNN	A*Net	0.987/98.4/98.7	0.967/94.8/99.1	0.549/49.5/65.9	0.411/32.1/58.6	0.549/48.6/65.2	0.563/49.8/68.6
GNN	AdaProp	0.988/98.6/99.0	0.969/95.6/99.5	0.562/49.9/67.1	0.417/33.1/58.5	0.554/49.3/65.5	0.573/51.0/68.5
GNN	one-shot-subgraph	0.988/98.7/99.0	0.972/95.5/99.4	0.567/51.4/66.6	0.304/22.3/45.4	0.547/48.5/65.1	0.606/54.0/72.1
GNN	MoKGR	0.993/99.1/99.3	0.978/96.5/99.6	0.611/53.9/70.2	0.443/36.8/60.7	0.584/50.7/67.9	0.657/57.7/75.8
GNN	**CERA-OSR**	**0.996/99.4/99.8**	**0.984/97.3/99.8**	**0.626/55.6/71.8**	**0.468/39.2/63.7**	**0.603/53.1/70.5**	**0.676/60.0/78.2**

Table 4. Component-removal results on the four core benchmarks. Avg. MRR is the unweighted mean, rounded to four decimals

Variant	WN18RR	FB15k-237	NELL-995	YAGO3-10	Avg. MRR
CERA-OSR full	0.626	0.468	0.603	0.676	0.5933
w/o depth selection	0.619	0.459	0.598	0.672	0.5870
w/o counterfactual training	0.607	0.446	0.592	0.665	0.5775
w/o reversible recovery	0.596	0.421	0.581	0.654	0.5630
w/o soft evidence router	0.574	0.338	0.558	0.617	0.5218
w/o RDG prior	0.568	0.304	0.547	0.606	0.5063
one-shot-subgraph backbone	0.567	0.304	0.547	0.606	0.5060

5.3. Evidence Faithfulness and Recovery

Table 5 evaluates gate AUC, edge-deletion sensitivity and recovery behavior. Removing top-gated edges reduces MRR by 0.159–0.183, compared with 0.020–0.026 for random-edge deletion. The larger decrease indicates that high-gated edges carry more predictive information under the evaluated perturbations.

Failure-to-Hits@10 transition rates range from 4.2% to 5.6%, while Hits@10-to-failure rates range from 1.2% to 1.8%. Recovery overhead ranges from 2.8% to 3.5%. These diagnostics characterize ranking changes and the additional evidence used during recovery.

Gate AUC ranges from 0.781 to 0.815 across the core benchmarks. Together with deletion sensitivity, these results characterize the association between learned evidence weights and predictive support. The analysis concerns model prediction fidelity under graph perturbations.

5.4. Counterfactual Robustness

Table 6 compares relation-swap gaps, path-corruption gaps, distractor MRR drops and top-deletion drops. On WN18RR, the relation-swap gap increases from 0.041 to 0.137 and the path-corruption gap from 0.036 to 0.128, while the distractor MRR drop decreases from 0.048 to 0.019.

Across the four datasets, CERA-OSR produces larger original-minus-corrupted score gaps and smaller distractor MRR drops than the one-shot-subgraph baseline. Score-gap magnitudes depend on the scoring scale; the MRR-drop columns provide complementary evidence based on ranking changes.

The results indicate sensitivity to the tested relation and path perturbations, together with reduced ranking degradation under distractor insertion.

5.5. Efficiency and Budget Analysis

Table 7 reports edge count, reopened edges, peak GPU memory and evaluation time. On WN18RR, edge count decreases from 18.9k to 13.1k, while evaluation time increases from 126.5 s to 150.0 s and memory from 12.4 GB to 15.5 GB. On NELL-995, edge count decreases from 34.5k to 24.8k, while time increases from 142.4 s to 185.0 s and memory from 18.6 GB to 22.0 GB.

These results show a trade-off between graph size and execution cost. Relation encoding, multiple depth experts and conditional second-pass reasoning introduce computation beyond the number of distinct edges. A smaller evidence graph can therefore require more memory and evaluation time.

On YAGO3-10, CERA-OSR requires 3300.0 s for evaluation, compared with the literature baseline value of 3723.7 s. This timing comparison uses a published baseline measurement, whereas the WN18RR and NELL-995 comparisons use local baseline measurements.

Table 5. Evidence sensitivity and recovery diagnostics. Deletion drops are absolute MRR reductions; recovery transitions and overhead are percentages

Dataset	Gate AUC	Top deletion	Random deletion	Wrong→H@10	H@10→Wrong	Overhead
WN18RR	0.792	0.183	0.026	4.8%	1.3%	+3.0%
FB15k-237	0.781	0.159	0.020	5.6%	1.8%	+3.5%
NELL-995	0.803	0.176	0.023	4.9%	1.5%	+3.1%
YAGO3-10	0.815	0.168	0.022	4.2%	1.2%	+2.8%

Table 6. Counterfactual robustness results. Score gaps are original–evidence scores minus corrupted–evidence scores; MRR drops are absolute reductions

Dataset	Method	Relation-swap gap	Path-corruption gap	Distractor MRR drop	Top deletion drop
WN18RR	one-shot-subgraph	0.041	0.036	0.048	0.061
WN18RR	CERA-OSR	0.137	0.128	0.019	0.183
FB15k-237	one-shot-subgraph	0.035	0.030	0.062	0.044
FB15k-237	CERA-OSR	0.121	0.113	0.024	0.159
NELL-995	one-shot-subgraph	0.039	0.033	0.055	0.057
NELL-995	CERA-OSR	0.132	0.120	0.021	0.176
YAGO3-10	one-shot-subgraph	0.038	0.032	0.059	0.052
YAGO3-10	CERA-OSR	0.126	0.117	0.020	0.168

Table 7. Efficiency measurements. Evaluation time is in seconds and peak GPU memory is in GB. Local and literature baseline rows are identified separately

Dataset	Method	MRR	Edges/query	Recovery edges	Peak GPU GB	Eval time
WN18RR	one-shot-subgraph local	0.5679	18.9k	0	12.4	126.5s
WN18RR	CERA-OSR	0.626	13.1k	390	15.5	150.0s
NELL-995	one-shot-subgraph local	0.5474	34.5k	0	18.6	142.4s
NELL-995	CERA-OSR	0.603	24.8k	610	22.0	185.0s
YAGO3-10	one-shot-subgraph reported	0.606	59.0k	0	31.0	3723.7s
YAGO3-10	CERA-OSR	0.676	42.0k	950	34.5	3300.0s

5.6. Long-Tail and Sparse Relations

Table 8 evaluates low-degree and rare-relation query groups on the four core benchmarks. CERA-OSR improves MRR over MoKGR by 0.025–0.038 across these groups, indicating benefits in settings with limited structural or relational support.

The gains on low-degree and rare-relation query groups complement the aggregate results by showing improved ranking performance when structural or relational support is limited.

6. Discussion, Limitations and Future Work

The ranking and ablation results support combining relation-aware retrieval with continuous evidence weighting and bounded recovery. Gates, recovered candidates and perturbation responses also make the selection process inspectable. Their influence on model predictions provides evidence of prediction fidelity, while establishing real-world causal explanations requires a different form of validation.

An illustrative urban workflow concerns the query (road segment, managed_by_agency, ?). A training-derived pool could include road–district, district–agency and service-ownership paths, plus unrelated high-degree transport links. Relation-conditioned gates would weight these candidates. If the first-pass agency scores are close, recovery would revisit low-gated ownership or administrative candidates already inside the pool.

An analyst could inspect the original and recovered paths, changes in candidate scores and the effect of removing selected edges. This illustrative scenario describes how the framework could support evidence inspection in an urban application. Evaluation with real urban entities, temporal records and analyst feedback remains future work.

The evaluation has several limitations. Five-seed means are reported without seed-wise dispersion or paired significance tests, so the numerical gains are interpreted descriptively. The ablation study isolates several components but does not separately remove

Table 8. Long-tail and sparse-relation results. Delta denotes the absolute MRR difference from MoKGR

Dataset	Bucket	one-shot MRR	MoKGR MRR	CERA-OSR MRR	Delta vs MoKGR
WN18RR	low-degree head/tail	0.492	0.548	0.574	+0.026
WN18RR	rare relation	0.421	0.476	0.512	+0.036
FB15k-237	low-degree head/tail	0.244	0.376	0.407	+0.031
FB15k-237	rare relation	0.198	0.331	0.369	+0.038
NELL-995	low-degree head/tail	0.493	0.536	0.566	+0.030
NELL-995	rare relation	0.456	0.502	0.538	+0.036
YAGO3-10	low-degree head/tail	0.551	0.612	0.637	+0.025
YAGO3-10	rare relation	0.503	0.575	0.609	+0.034

the complete multi-view pool. Further controlled comparisons would strengthen assessment of training variability and component interactions.

The benchmarks are general knowledge graphs, so the results do not establish performance on urban tasks. Evaluation on resources such as UUKG [13] requires attention to data provenance, temporal validity and task-specific interpretation, consistent with the contextual assessment of trustworthy AI [18].

Computational cost is another limitation. Multiple depth experts and conditional second-pass reasoning can increase memory and latency, as shown on WN18RR and NELL-995. Recovery is also bounded by initial-pool recall. Counterfactual training depends on corruption quality, since an unobserved true fact can be treated incorrectly as a negative example.

Future work will investigate paired statistical analysis, a complete multi-view-pool ablation, matched computational comparisons and evaluation on a real urban knowledge graph. Calibrated uncertainty and temporal evidence are further directions for improving recovery and application-specific reliability.

7. Conclusion

CERA-OSR integrates train-only multi-view pooling, relation-conditioned soft gates, multi-depth reasoning, bounded recovery and counterfactual training into a one-shot subgraph reasoning framework. It reconsiders low-gated candidates under uncertain rankings and exposes the evidence involved in the resulting predictions.

On the four core benchmarks, CERA-OSR achieves a mean MRR of 0.5933, an absolute improvement of 0.0195 over the MoKGR literature mean. Ablation, deletion and counterfactual analyses characterize component contributions and prediction sensitivity, while efficiency measurements reveal the computational trade-off of recovery. Future evaluation will address statistical robustness and application to urban knowledge graphs.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability

The benchmark datasets are publicly available from their original providers and repositories. The model configuration, random seeds and evaluation protocol are described in Section 4.

Acknowledgements

The authors thank the research groups that released benchmark datasets, baseline implementations and evaluation protocols for knowledge graph reasoning. The authors also acknowledge the community contributions behind one-shot subgraph reasoning, adaptive graph reasoning and mixture-based KG reasoning, which provide the basis for comparison and extension in this work.

References

[1] Dettmers T, Minervini P, Stenetorp P, Riedel S. Convolutional 2D Knowledge Graph Embeddings. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence. AAAI Press; 2018. p. 1811–1818. doi:10.1609/aaai.v32i1.11573.

[2] Sun Z, Deng ZH, Nie JY, Tang J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In: International Conference on Learning Representations. 2019. Available from: https://openreview.net/forum?id=HkgEQnRqYQ.

[3] Zhang S, Tay Y, Yao L, Liu Q. Quaternion Knowledge Graph Embeddings. In: Advances in Neural Information Processing Systems. Vol. 32. Curran Associates; 2019. Available from: https://proceedings.neurips.cc/paper/2019/hash/d961e9f236177d65d21100592edb0769-Abstract.html.

[4] Das R, Dhuliawala S, Zaheer M, Vilnis L, Durugkar I, Krishnamurthy A, Smola A, McCallum A. Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning. In: International Conference on Learning Representations. 2018. Available from: https://openreview.net/forum?id=Syg-YfWCW.

[5] Sadeghian A, Armandpour M, Ding P, Wang DZ. DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs. In: Advances in Neural Information Processing Systems. Vol. 32.

Curran Associates; 2019. Available from: https://proceedings.neurips.cc/paper/2019/hash/0c72cb7ee1512f800abe27823a792d03-Abstract.html.

[6] Vashishth S, Sanyal S, Nitin V, Talukdar P. Composition-based Multi-Relational Graph Convolutional Networks. In: International Conference on Learning Representations. 2020. Available from: https://openreview.net/forum?id=BylA_C4tPr.

[7] Zhu Z, Zhang Z, Xhonneux LP, Tang J. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. In: Advances in Neural Information Processing Systems. Vol. 34. Curran Associates; 2021. p. 29476–29490. Available from: https://proceedings.neurips.cc/paper/2021/hash/f6a673f09493afcd8b129a0bcf1cd5bc-Abstract.html.

[8] Zhang Y, Yao Q. Knowledge Graph Reasoning with Relational Digraph. In: Proceedings of the ACM Web Conference 2022. ACM; 2022. p. 912–924. doi:10.1145/3485447.3512008.

[9] Zhang Y, Zhou Z, Yao Q, Chu X, Han B. AdaProp: Learning Adaptive Propagation for Graph Neural Network based Knowledge Graph Reasoning. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM; 2023. p. 3446–3457. doi:10.1145/3580305.3599404.

[10] Zhu Z, Yuan X, Galkin M, Xhonneux LP, Zhang M, Gazeau M, Tang J. A*Net: A Scalable Path-based Reasoning Approach for Knowledge Graphs. In: Advances in Neural Information Processing Systems. Vol. 36. Curran Associates; 2023. p. 59323–59336. doi:10.52202/075280-2591.

[11] Zhou Z, Zhang Y, Yao J, Yao Q, Han B. Less is More: One-shot Subgraph Reasoning on Large-scale Knowledge Graphs. In: International Conference on Learning Representations. 2024. Available from: https://proceedings.iclr.cc/paper_files/paper/2024/hash/c3eb94d149aea08c28505d1e5234a21a-Abstract-Conference.html.

[12] Du E, Liu S, Zhang Y. Mixture of Length and Pruning Experts for Knowledge Graphs Reasoning. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics; 2025. p. 432–453. doi:10.18653/v1/2025.emnlp-main.23.

[13] Ning Y, Liu H, Wang H, Zeng Z, Xiong H. UUKG: Unified Urban Knowledge Graph Dataset for Urban Spatiotemporal Prediction. In: Advances in Neural Information Processing Systems. Vol. 36, Datasets and Benchmarks Track. Curran Associates; 2023. p. 62442–62456. doi:10.52202/075280-2727.

[14] Meilicke C, Chekol MW, Ruffinelli D, Stuckenschmidt H. Anytime Bottom-Up Rule Learning for Knowledge Graph Completion. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. IJCAI; 2019. p. 3137–3143. doi:10.24963/ijcai.2019/435.

[15] Zhang S, Zhang J, Song X, Adeshina S, Zheng D, Faloutsos C, Sun Y. PaGE-Link: Path-based Graph Neural Network Explanation for Heterogeneous Link Prediction. In: Proceedings of the ACM Web Conference 2023. ACM; 2023. p. 3784–3793. doi:10.1145/3543507.3583511.

[16] Amara K, Ying Z, Zhang Z, Han Z, Zhao Y, Shan Y, Brandes U, Schemm S, Zhang C. GraphFramEx: Towards Systematic Evaluation of Explainability Methods for Graph Neural Networks. In: Proceedings of the First Learning on Graphs Conference. Proceedings of Machine Learning Research. Vol. 198. PMLR; 2022. p. 44:1–44:23. Available from: https://proceedings.mlr.press/v198/amara22a.html.

[17] Ning Y, Liu H. UrbanKGent: A Unified Large Language Model Agent Framework for Urban Knowledge Graph Construction. In: Advances in Neural Information Processing Systems. Vol. 37. Curran Associates; 2024. p. 123127–123154. doi:10.52202/079017-3913.

[18] Tabassi E. Artificial Intelligence Risk Management Framework (AI RMF 1.0). Gaithersburg (MD): National Institute of Standards and Technology; 2023. Report No.: NIST AI 100-1. doi:10.6028/NIST.AI.100-1.

[19] Toutanova K, Chen D. Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality. Association for Computational Linguistics; 2015. p. 57–66. doi:10.18653/v1/W15-4007.

[20] Mahdisoltani F, Biega J, Suchanek FM. YAGO3: A Knowledge Base from Multilingual Wikipedias. In: Proceedings of the 7th Biennial Conference on Innovative Data Systems Research (CIDR). 2015. Available from: https://www.cidrdb.org/cidr2015/Papers/CIDR15_Paper1.pdf.