Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Framework for Dynamic Knowledge Graph Ontology Incremental Alignment based on Large Language Model Agents

Hao Qin
Guangdong Power Grid Company Limited
Fusheng Wei
Guangdong Power Grid Company Limited
Bin Liang
gddwjczx@163.com

Guangdong Power Grid Company Limited
Wobin Lin
Guangdong Power Grid Company Limited

Article

Keywords:

Posted Date: July 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9503150/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A Framework for Dynamic Knowledge Graph Ontology Incremental Alignment based on Large Language Model Agents

Hao Qin[1], **Fusheng Wei**[1], **Bin Liang**[1,*], **and Wobin Lin**[1]

[1]Guangdong Power Grid Company Limited, Guangzhou 510600, China
[*]gddwjczx@163.com

ABSTRACT

Dynamic knowledge graphs used in risk management and compliance monitoring evolve continuously, yet most ontology alignment methods still operate on static snapshots, which creates a persistent trade-off between semantic precision and update latency. To address this limitation, we propose DOKA-Agent, a large language model agent framework for dynamic ontology incremental alignment. The framework follows a perceive-plan-act workflow: it captures structural and content deltas from a graph, enriches the semantic signature of affected entities using contextual information, retrieves candidates through a hybrid dense-sparse strategy, and then produces relation-aware alignment decisions with confidence estimation and constraint checking. Experiments on DBP15K dynamic partitions show that DOKA-Agent improves Hit@1, Hit@10, and F1 in both static and dynamic settings while substantially reducing alignment time. These results indicate that agentic orchestration provides a practical way to maintain semantic consistency in continuously evolving knowledge graphs.

Introduction

Knowledge graphs have become an important infrastructure for intelligent decision support in domains such as recommendation, finance, industrial safety, education, and legal document analysis[1–4]. In practical deployments, however, these graphs are rarely static. New entities, revised rules, evolving domain knowledge, and continuously arriving events reshape the graph over time, making semantic maintenance a moving target[5–7]. This evolution introduces term redundancy, concept drift, and alignment inconsistency, all of which can directly reduce the reliability of downstream retrieval, reasoning, and decision-making[8,9]. At the same time, recent graph-based intelligent systems have shown that preserving long-range structural dependencies and multi-scale semantics is essential when modelling complex relationships[10–12], which further motivates structure-aware alignment in dynamic environments[13–15].

Ontology alignment is a core mechanism for resolving semantic heterogeneity across knowledge graphs, but conventional methods remain limited in dynamic scenarios. Static alignment methods usually require re-running the alignment pipeline after each update, which is costly and poorly suited to streaming environments[16,17]. Traditional incremental approaches reduce recomputation, yet they mainly focus on structural updates and often struggle to interpret rich textual descriptions or semantic changes in context[7,18–21]. Large language models have improved semantic matching for ontology alignment[22–27], but most LLM-based methods are still designed for static or batch alignment rather than persistent graph evolution.

This gap motivates an agent-based formulation. Instead of treating alignment as a single monolithic prediction step, we frame it as a dynamic workflow in which a large language model agent perceives graph changes, plans tool use, retrieves candidates, evaluates multi-view similarity, and commits validated ontology relations. This paper makes three contributions. First, it proposes DOKA-Agent, a dynamic ontology alignment framework tailored to evolving knowledge graphs. Second, it designs an incremental perceive-plan-act workflow that integrates context completion, hybrid retrieval, and relation-aware decision-making. Third, it validates the framework on dynamic knowledge graph benchmarks and shows that it improves both alignment quality and operational efficiency.

Related work

Ontology alignment methods

Ontology alignment methods broadly include string-based, embedding-based, and hybrid approaches. String-based systems remain attractive because of their interpretability and efficiency. POMap improves pairwise ontology matching by optimizing string metrics and thresholds[28], while semantic-association and probabilistic-logic methods further enhance repair and consistency control[29]. Hybrid approaches combine lexical cues with learned semantics. For example, fuzzy string matching

combined with machine learning improves robustness to noisy labels and near-duplicate concepts[30], and FLORA uses fuzzy logic to support unsupervised alignment with dangling entities[31].

Embedding-based methods map entities and relations into continuous spaces. Entity-pair embedding methods model alignment through learned representation similarity[32]. Graph neural approaches enhance this process by incorporating neighborhood structure and relation patterns[16,33,34]. Contextual encoder methods such as BERTMap introduce stronger text understanding for ontology matching[23], and recent work explores using generated descriptions to improve candidate retrieval and disambiguation[26]. Despite these advances, cost and responsiveness remain major issues in continuously changing graphs[27].

Pre-trained language models for ontology alignment

Pre-trained language models have substantially improved static ontology alignment by learning dense contextual representations that capture synonymy, paraphrase, and cross-lingual correspondence[22,24]. Quantized encoder pipelines can improve inference efficiency in specialized biomedical settings[35], while multimodal contrastive methods extend entity alignment beyond pure text[36]. In domain-specific settings such as labor-market alignment, large language models combined with domain knowledge can improve concept matching and refinement[37].

Generative LLMs further enable zero-shot and few-shot alignment through prompting. OLaLa and related prompt-based approaches show that LLMs can reason over ontology labels and descriptions without task-specific retraining[24,25]. Other studies investigate prefix design for non-English alignment and benchmark hallucination behavior in ontology matching[38,39]. More recent work also studies LLM-driven ontology matching with explicit search procedures and depth-first prioritization[27]. These results are promising, but they still leave open the problem of how to integrate LLM reasoning with continuous graph updates, external tools, and persistent alignment state.

Dynamic and evolving knowledge graphs

Dynamic knowledge graph research has mainly emphasized efficient updating. Time-aware embedding methods model temporal evolution directly in entity representations[6]. Reinforcement learning and continual learning have also been explored for dynamic reasoning and evolving knowledge retention[7,20]. Incremental and scale-aware evolution strategies reduce the cost of retraining by focusing on changed graph regions or gradually expanding representation capacity[18,19,21]. Snapshot and streaming graph learning methods address large-scale dynamic processing by balancing temporal segmentation, throughput, and latency[40–43]. However, many of these methods prioritize efficiency over semantic reconstruction, and few explicitly target ontology incremental alignment under rich semantic drift.

Methods

DOKA-Agent is an LLM-agent framework for dynamic ontology incremental alignment. Its goal is to maintain semantic coherence in a continuously evolving knowledge graph while avoiding the high cost of repeated full-graph alignment. The framework is organized around three modules: Dynamic Perception, Agent Decision-Making Core, and Tools and Knowledge Base. Figure 1 illustrates the overall workflow.

Dynamic perception module

The Dynamic Perception module is responsible for near-lossless capture and structured transformation of graph updates. Its input is a stream of raw database events,

$$E_{\text{raw}} = \{E_1, E_2, \ldots, E_n\}, \tag{1}$$

where each event contains operation metadata, timestamps, and changed graph objects. The module translates these low-level operations into alignment-relevant semantic events such as entity creation, deletion, relation modification, or property update, and serializes them into structured task descriptions:

$$T = \text{Format}(E) = (e_{\text{id}}, t, s, e_{\text{type}}, \text{Struct}(e_{\text{details}})). \tag{2}$$

To measure capture quality, we define event coverage as

$$\text{Coverage} = \frac{|E_{\text{captured}}|}{|E_{\text{total}}|}, \tag{3}$$

where E_{captured} is the set of events collected by the listener and E_{total} is the total number of relevant changes within a time window. We further evaluate task completeness:

$$\text{Completeness}(T) = \frac{|F_{\text{filled}}(T)|}{|F_{\text{required}}(e_{\text{type}})|}. \tag{4}$$

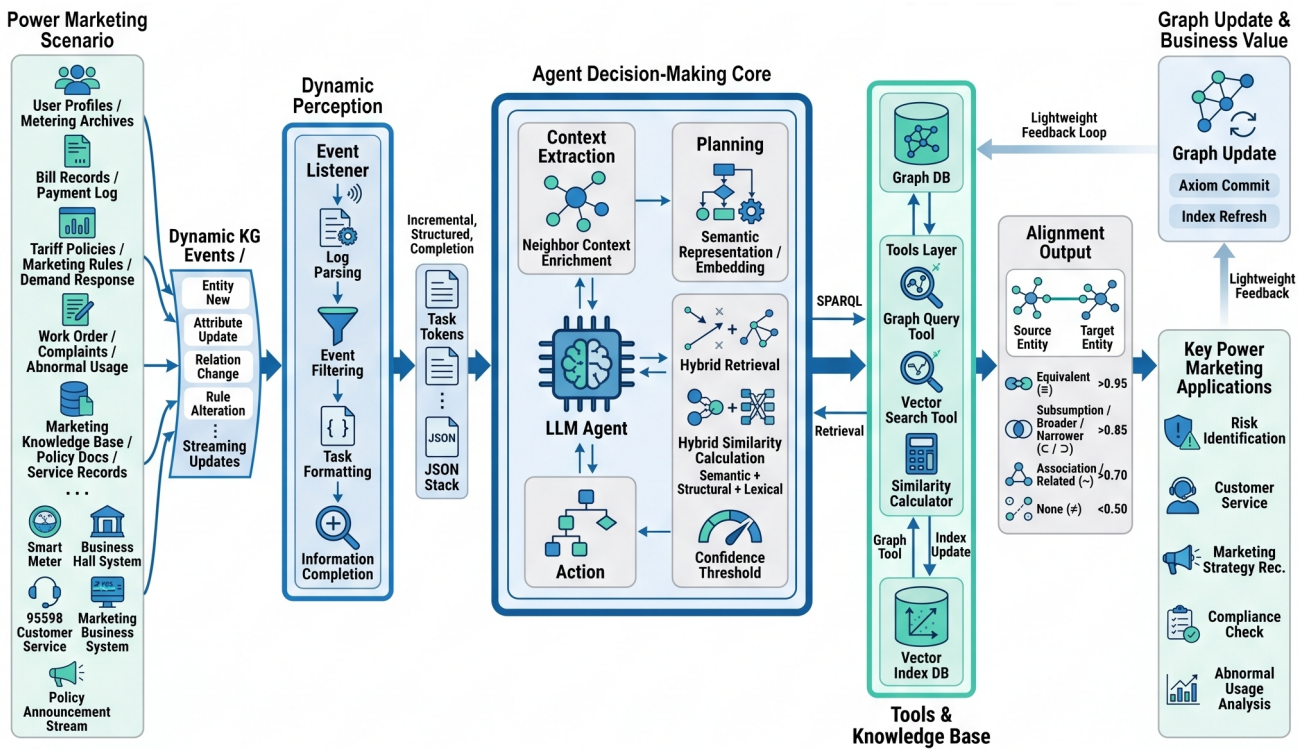


Figure 1. Overall workflow of DOKA-Agent. The framework follows a perceive-plan-act loop in which graph changes are captured, transformed into structured tasks, processed by an LLM agent with external tools, and committed back to the graph as validated ontology relations.

If completeness falls below a threshold, the module invokes a graph query tool to supplement missing attributes or neighborhood context before forwarding the task to the agent.

Agent decision-making core

The Agent Decision-Making Core receives a structured task T, a toolset

$$Tool = \{\text{graph_query_tool}, \text{vector_search_tool}, \text{similarity_calculator}\},$$

and a knowledge index database. It outputs an alignment result

$$R = \{r, \theta\},$$

where $r \in \{\text{equivalent}, \text{broader}, \text{narrower}, \text{related}, \text{none}\}$ denotes the predicted relation type and $\theta \in [0,1]$ is the confidence.

Planning phase

The planning phase decomposes each incoming task into executable subtasks:

$$P = \text{Decompose}(T, \text{LLM}) = [p_1, p_2, \ldots, p_k]. \tag{5}$$

This step allows the agent to adapt its tool-invocation order to event type and semantic complexity. The relevance of the generated plan is estimated by

$$\text{Validity}(P,T) = \frac{|P_{\text{relevant}}(P,T)|}{k}. \tag{6}$$

The agent then queries graph context for the affected entity:

$$C = \text{QueryGraph}(T.e_{\text{details}}.\text{entity_id}, T.e_{\text{details}}.\text{entity_type}, \text{graph_query_tool}), \tag{7}$$

where $C = (d, N, \mathscr{R})$ contains descriptive text, neighboring entities, and relations. We characterize semantic richness through

$$\text{Density}(C) = \frac{\text{Len}(d) + \sum_{n \in N} \text{Len}(n_{\text{desc}})}{1 + |N|}. \tag{8}$$

Action phase

The action phase first generates a unified semantic description and embedding:

$$(d', v) = \text{LLM_Encode}(C, \text{LLM}), \tag{9}$$

followed by normalization,

$$v_{\text{norm}} = \frac{v}{\|v\|_2}. \tag{10}$$

Candidate entities are then retrieved from the vector index:

$$\mathscr{S} = \text{Retrieve}(v_{\text{norm}}, \text{IndexDB}, \text{vector_search_tool}, K). \tag{11}$$

For each candidate s_i, DOKA-Agent computes a hybrid similarity score:

$$\begin{aligned} \text{Sim}(v_{\text{norm}}, s_i) = {} & w_1 \cdot \text{Semantic_Sim}(v_{\text{norm}}, s_v) \\ & + w_2 \cdot \text{Structural_Sim}(C, s_{C,i}) \\ & + w_3 \cdot \text{Lexical_Sim}(d', s_d), \end{aligned} \tag{12}$$

where $w_1 + w_2 + w_3 = 1$ and the default weights are 0.5, 0.3, and 0.2, respectively. Structural similarity is defined as

$$\text{Structural_Sim}(C, s_{C,i}) = \frac{|\mathscr{R} \cap s_{\mathscr{R},i}|}{|\mathscr{R} \cup s_{\mathscr{R},i}|}, \tag{13}$$

and lexical similarity is estimated through normalized edit distance:

$$\text{Lexical_Sim}(d', s_d) = 1 - \frac{\text{EditDistance}(d', s_d)}{\max(\text{Len}(d'), \text{Len}(s_d))}. \tag{14}$$

The final relation for each candidate is produced by the LLM:

$$(r_i, \theta_i) = \text{LLM_Decide}(C, s_{C,i}, \text{Sim}(v_{\text{norm}}, s_i), \text{LLM}). \tag{15}$$

The system selects the highest-confidence decision,

$$(r, \theta) = \arg\max_{i=1\ldots K} \theta_i, \tag{16}$$

and rejects low-confidence alignments when $\theta < \theta_{\text{thresh}}$ with $\theta_{\text{thresh}} = 0.7$.

Tools and knowledge base

The tool layer provides external computation and storage support. The Graph Query Tool extracts neighborhood context from the graph database and is evaluated by mean query latency:

$$\text{Latency}_{\text{query}} = \frac{\sum_{i=1}^{n} t_i}{n}. \tag{17}$$

The Vector Search Tool performs efficient similarity search over an incremental vector index, and its throughput is measured as

$$\text{Throughput}_{\text{search}} = \frac{m}{T}, \tag{18}$$

where m is the number of search requests completed during time window T. Together, the Graph DB and IndexDB form a dual-layer knowledge base that supports both semantic retrieval and persistent graph updates.

Algorithm optimization

To optimize the hybrid similarity weights, we use a total loss that combines cross-entropy and L2 regularization:

$$L_{\text{total}} = L_{\text{ce}} + \lambda L_{\text{reg}}, \tag{19}$$

where $\lambda = 0.01$. The cross-entropy loss is computed over annotated alignment decisions:

$$L_{\text{ce}} = -\frac{1}{M}\sum_{j=1}^{M}\left[\theta_{j,\text{true}}\log(\theta_{j,\text{pred}}) + (1-\theta_{j,\text{true}})\log(1-\theta_{j,\text{pred}})\right]. \tag{20}$$

The regularizer is

$$L_{\text{reg}} = \frac{1}{3}\sum_{k=1}^{3} w_k^2. \tag{21}$$

The derivative for each weight is

$$\frac{\partial L_{\text{total}}}{\partial w_k} = \frac{\partial L_{\text{ce}}}{\partial w_k} + \lambda \cdot \frac{2}{3} w_k. \tag{22}$$

Algorithm 1 Condensed DOKA-Agent alignment process

while event listener is active **do**
  capture raw graph event and format task T
  if Completeness$(T) < 0.9$ **then**
    supplement missing context via graph query
  end if
  decompose T into plan P
  query entity context C
  encode C into (d', v) and normalize v_{norm}
  retrieve top-K candidates $\mathscr{S}$
  for all $s_i \in \mathscr{S}$ **do**
    compute hybrid similarity $\text{Sim}(v_{\text{norm}}, s_i)$
    infer relation (r_i, θ_i)
  end for
  select highest-confidence result (r, θ)
  if $\theta \geq \theta_{\text{thresh}}$ **then**
    commit aligned relation to graph database
  end if
  update embedding index and logs
end while

Results

Experimental setup

To simulate graph evolution, the gold alignments were partitioned into three time steps, t_0, t_1, and t_2. At t_0, 12,000 alignments were retained to form the initial graph. At t_1 and t_2, 1,500 additional alignment pairs were introduced at each step together with their associated entities and edges, gradually reconstructing the original benchmark.

For static evaluation, the complete set of 15,000 alignments was split into 30% training and 70% testing. For dynamic evaluation, the train-test split was performed within each time step. Newly introduced alignments at t_1 and t_2 were used for incremental training, while the test set accumulated historical test instances from earlier time steps. This setting follows common dynamic alignment protocols and allows us to evaluate both static performance and incremental adaptation.

Baselines and metrics

We compared DOKA-Agent against four categories of methods. Static full-graph baselines included DINGAL-B, HGCN-JE[44], and LLM-Static. Dynamic incremental baselines included DINGAL-O, DINGAL-U, and BootEA. Traditional alignment baselines included MTransE[45] and RDGCN[46]. We also included an LLM-Direct baseline that uses a large language model without explicit planning or tool orchestration.

Alignment effectiveness was measured using Hit@1, Hit@10, Precision, Recall, and F1-score. Hit@k was computed from the ranked candidate list without thresholding. Precision, Recall, and F1 used a confidence threshold of 0.7, consistent with the decision rule defined in the method. Operational efficiency was assessed using average processing time per change event and total alignment time per dynamic step.

Static alignment performance

Table 1 shows that DOKA-Agent achieved the best overall performance on all three DBP15K subsets. Compared with DINGAL-B, it improved Hit@1 and F1 consistently, indicating that the agent-based pipeline contributes not only to dynamic responsiveness but also to one-shot alignment quality. The gains reflect the benefit of combining semantic encoding, structural context, and lexical correction within a unified decision process. Compared with pure LLM retrieval, the tool-assisted design also reduces candidate noise and improves the quality of final ranking.

Table 1. Static alignment performance comparison on DBP15K.

Method	DBP15K-ZH-EN			DBP15K-JA-EN			DBP15K-FR-EN		
	Hit@1 (%)	Hit@10 (%)	F1 (%)	Hit@1 (%)	Hit@10 (%)	F1 (%)	Hit@1 (%)	Hit@10 (%)	F1 (%)
MTransE	30.8	61.4	42.1	27.9	57.5	38.6	24.4	55.6	35.2
RDGCN	70.8	84.6	77.2	76.7	89.5	82.6	88.6	95.7	91.9
HGCN-JE	72.0	85.7	78.3	76.6	89.7	82.8	89.2	96.1	92.5
DINGAL-B	72.5	87.4	79.1	78.2	91.3	84.3	92.0	97.9	94.8
LLM-Static (GPT-4)	73.1	88.0	79.8	78.9	91.8	84.9	92.6	98.2	95.3
DOKA-Agent	**73.9**	**88.8**	**80.6**	**79.7**	**92.6**	**85.7**	**93.5**	**99.0**	**96.2**

Dynamic alignment performance

The dynamic evaluation at t_1 and t_2 further highlights the value of the perceive-plan-act workflow. As shown in Table 2, DOKA-Agent consistently outperformed DINGAL-O, DINGAL-U, and LLM-Direct. These results suggest that explicitly capturing change events, retrieving targeted candidates, and re-estimating relations with updated context is more effective than relying on either purely structural incremental updates or direct LLM matching alone[17]. The advantage becomes more pronounced in later time steps, where semantic drift and accumulated graph changes make naive local updates less reliable.

Table 2. Dynamic alignment performance comparison.

Method	DBP15K-ZH-EN			DBP15K-JA-EN			DBP15K-FR-EN		
	Hit@1 (%)	Hit@10 (%)	F1 (%)	Hit@1 (%)	Hit@10 (%)	F1 (%)	Hit@1 (%)	Hit@10 (%)	F1 (%)
BootEA	44.1	76.9	55.7	44.8	78.5	56.6	45.2	80.8	57.3
DINGAL-O	71.6	86.8	78.5	77.3	90.3	83.3	91.8	97.6	94.5
DINGAL-U	71.7	87.5	78.8	77.4	90.6	83.5	91.4	97.5	94.3
LLM-Direct (GPT-4)	72.3	88.1	79.4	78.1	91.2	84.1	92.1	98.0	94.9
DOKA-Agent	**72.9**	**88.6**	**80.0**	**78.8**	**91.9**	**84.9**	**93.2**	**98.7**	**95.8**

Operational efficiency

Operational efficiency is critical for live graph maintenance. On DBP15K-ZH-EN at t_1, DOKA-Agent achieved an average processing time of 0.9 s and a total alignment time of 3.8 s, outperforming static recomputation and remaining faster than other dynamic baselines. The efficiency gain stems from three design choices: event-level perception avoids full scans, vector retrieval narrows the candidate space before detailed reasoning, and incremental knowledge-base updates prevent repeated recomputation of unchanged embeddings. This pattern is consistent with broader observations in dynamic graph learning, where indexing and local updates dominate end-to-end responsiveness[41,42,47].

Table 3. Dynamic running efficiency comparison on DBP15K-ZH-EN.

Method	Average Processing Time (s)	Total Alignment Time (s)	Efficiency Improvement vs DINGAL-B (%)
DINGAL-B	89.2	102.5	-
DINGAL-O	1.0	4.2	95.9
DINGAL-U	9.8	12.6	87.7
LLM-Direct (GPT-4)	2.3	8.5	91.7
DOKA-Agent	**0.9**	**3.8**	**96.3**

Case study, ablation, and sensitivity

A representative example at t_1 involves the cross-lingual entity pair "Alibaba Group". Baseline methods failed to rank the correct counterpart first because their decisions were dominated either by incomplete local structure or by purely textual comparison. In contrast, DOKA-Agent used the perception module to capture the update event, the graph query tool to enrich missing neighbors, and the hybrid similarity function to combine semantic, structural, and lexical evidence. This resulted in a higher-confidence correct match and lower retrieval time.

The ablation study in Table 4 shows that each core component contributes to performance. Removing the planning module reduces adaptability to event type, removing semantic similarity most strongly harms alignment quality, and removing information completion reduces the quality of context used for decision-making. The weight sensitivity analysis further indicates that the default configuration of semantic, structural, and lexical weights balances performance well across multilingual settings.

Table 4. DOKA-Agent ablation study on DBP15K-FR-EN.

Model Variant	Hit@1 (%)	Hit@10 (%)	F1 (%)
DOKA-Agent (Complete)	93.5	99.0	96.2
Without Planning Module	91.2	97.8	94.4
Without Semantic Similarity Weight	90.4	97.2	93.6
Without Information Completion	92.0	98.3	94.9

Discussion

The results indicate that dynamic ontology alignment benefits from agentic orchestration rather than a single-stage predictor. DOKA-Agent combines semantic reasoning from LLMs with targeted tool use and persistent graph-state management, which helps it respond to streaming changes without sacrificing alignment quality. This design also creates a flexible interface for future extensions, such as stronger constraint checking, cross-lingual prompting strategies[38,48], and more specialized domain adaptation for sectors such as industrial knowledge graphs, education, and safety management[3,5,16].

Recent studies also suggest that agentic and LLM-guided pipelines are becoming useful well beyond ontology alignment, including balanced bimodal guidance for visual generation[49], professional-grade power-system optimization modelling and safe energy-system reconfiguration[50,51], anonymous zero-shot fraud detection[52], multimodal affect perception and conversational emotion recognition[53,54], and hallucination-aware preference optimization for perceptual reasoning[55]. These adjacent advances reinforce the broader feasibility of combining foundation models with domain tools, retrieval, and calibrated decision procedures.

There are still limitations. The current implementation depends on the quality of graph-side event capture and on the stability of LLM outputs. LLM-based ontology matching is known to be vulnerable to hallucination and prompt sensitivity[24,39]. In addition, the present evaluation focuses on DBP15K, which does not fully cover domain-specific ontology constraints or real-world industrial update noise. Future work should explore multi-agent collaboration, domain-specific validators, and larger streaming benchmarks, while also drawing on recent progress in multimodal and domain-specific LLM systems[49–51,53].

References

1. Deepak, G., Ahmed, A. & Skanda, B. An intelligent inventive system for personalised webpage recommendation based on ontology semantics. *Int. J. Intell. Syst. Technol. Appl.* **18**, 115–132 (2019).
2. Mohnish, S., Deepak, G. & Pushpa, C. N. Ciskgg: A collective intelligence strategy for heterogeneous knowledge graph generation for criminology as a domain of choice. *2022 IEEE Int. Conf. for Women Innov. Technol. & Entrepreneurship (ICWITE)* 1–10 (2022).
3. Lin, L. Construction of intelligent educational teaching evaluation system based on knowledge graph and natural language processing. *2024 IEEE 2nd Int. Conf. on Sensors, Electron. Comput. Eng. (ICSECE)* 956–960 (2024).
4. Han, L. Research on knowledge graph construction technology based on intellectual property legal documents. *J. Intell. Knowl. Eng.* (2024).
5. Zhao, Q. & Liu, C. Construction and dynamic evolution of an ideological and political knowledge graph via event logic reasoning. *2025 4th Int. Conf. on Electron. Inf. Technol. (EIT)* 620–626 (2025).

6. Tang, X. *et al.* Timespan-aware dynamic knowledge graph embedding by incorporating temporal evolution. *IEEE Access* **8**, 6849–6860 (2020).

7. Liu, H. *et al.* Dynamic knowledge graph reasoning based on deep reinforcement learning. *Knowledge-Based Syst.* **241**, 108235 (2022).

8. Li, Y. *et al.* Sage: Scale-aware gradual evolution for continual knowledge graph embedding. *Proc. 31st ACM SIGKDD Conf. on Knowl. Discov. Data Min. V.2* (2025).

9. Liu, J.-C., Chen, C.-T., Lee, C. & Huang, S.-H. Evolving knowledge graph representation learning with multiple attention strategies for citation recommendation system. *ACM Transactions on Intell. Syst. Technol.* **15**, 1–26 (2024).

10. Cai, M., Chen, C. L. P., Li, S. & Zhang, T. Label feature co-learning for facial and eeg emotion recognition. *IEEE Transactions on Affect. Comput.* **16**, 3421–3433, DOI: 10.1109/TAFFC.2025.3600267 (2025).

11. Li, Q., Zhang, T., Chen, C. L. P., Yi, K. & Chen, L. Residual GCB-Net: Residual graph convolutional broad network on emotion recognition. *IEEE Transactions on Cogn. Dev. Syst.* **15**, 1673–1685, DOI: 10.1109/TCDS.2022.3147839 (2023).

12. Li, Q., Zhang, T., Chen, C. L. P., Zhang, X. & Hu, B. DGC-Link: Dual-gate chebyshev linkage network on eeg emotion recognition. *IEEE Transactions on Affect. Comput.* **16**, 3499–3511, DOI: 10.1109/TAFFC.2025.3602823 (2025).

13. Zhang, T., Li, Q., Yang, X. & Chen, C. L. P. MCGC-Net: Multi-scale controllable graph convolutional network on music emotion recognition. *IEEE Transactions on Affect. Comput.* **17**, 162–175, DOI: 10.1109/TAFFC.2025.3608830 (2026).

14. Li, Q., Chen, C. & Zhang, T. Memo2496: Expert-annotated dataset and dual-view adaptive framework for music emotion recognition. *arXiv preprint arXiv:2512.13998* (2025).

15. Zhang, T., Li, Q., Wen, J. & Chen, C. L. P. Enhancement and optimisation of human pose estimation with multi-scale spatial attention and adversarial data augmentation. *Inf. Fusion* **111**, 102522, DOI: 10.1016/j.inffus.2024.102522 (2024).

16. Wu, S. *et al.* Concept-aware entity alignment network for industrial knowledge graph. *IEEE Transactions on Ind. Informatics* **21**, 4316–4323 (2025).

17. Yan, Y., Liu, L., Ban, Y., Jing, B. & Tong, H. Dynamic knowledge graph alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 4564–4572 (2021).

18. Ding, Z. *et al.* A simple but powerful graph encoder for temporal knowledge graph completion. In *Intelligent Systems Conference*, 729–747 (2023).

19. Liu, J. *et al.* Towards continual knowledge graph embedding via incremental distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 8759–8768 (2024).

20. Daruna, A., Gupta, M., Sridharan, M. & Chernova, S. Continual learning of knowledge graph embeddings. *IEEE Robotics Autom. Lett.* **6**, 1128–1135 (2021).

21. Li, Y. *et al.* Sage: Scale-aware gradual evolution for continual knowledge graph embedding. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1600–1611 (2025).

22. Schneider, F., Dash, S., Bagchi, S., Mihindukulasooriya, N. & Gliozzo, A. M. Natural language focused ontology alignment. In *Proceedings of the 12th Knowledge Capture Conference 2023*, 114–121 (2023).

23. He, Y., Chen, J., Antonyrajah, D. & Horrocks, I. Bertmap: a bert-based ontology alignment system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 5684–5691 (2022).

24. He, Y., Chen, J., Dong, H. & Horrocks, I. Exploring large language models for ontology alignment. *arXiv preprint arXiv:2309.07172* (2023).

25. Hertling, S. & Paulheim, H. Olala: Ontology matching with large language models. In *Proceedings of the 12th Knowledge Capture Conference 2023*, 131–139 (2023).

26. Song, Y., Chen, J. & Schmidt, R. A. Genom: Ontology matching with description generation and large language model. *arXiv preprint arXiv:2508.10703* (2025).

27. Taboada, M., Martinez, D., Arideh, M. & Mosquera, R. Ontology matching with large language models and prioritized depth-first search. *Inf. Fusion* 103254 (2025).

28. Laadhar, A. *et al.* Pomap: An effective pairwise ontology matching system. In *IC3K 2017: 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 161–168 (2017).

29. Wu, N., Lai, X., Chen, M. & Pan, Y. Ontology matching and repair based on semantic association and probabilistic logic. *IEICE Transactions on Inf. Syst.* **107**, 1433–1443 (2024).

30. Rudwan, M. S. M. & Fonou-Dombeu, J. V. Hybridizing fuzzy string matching and machine learning for improved ontology alignment. *Futur. Internet* **15**, 229 (2023).

31. Peng, Y., Bonald, T. & Suchanek, F. M. Flora: Unsupervised knowledge graph alignment by fuzzy logic. In *International Semantic Web Conference*, 196–215 (2025).

32. Wang, Z., Yang, J. & Ye, X. Knowledge graph alignment with entity-pair embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1672–1680 (2020).

33. Sakong, D. *et al.* Complex representation learning with graph convolutional networks for knowledge graph alignment. *Int. J. Intell. Syst.* **2023**, 5494961 (2023).

34. Xiang, Y. *et al.* Ontoea: Ontology-guided entity alignment via joint knowledge graph embedding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 1117–1128 (2021).

35. Bouaggad, O. & Grabar, N. Search-optimized quantization in biomedical ontology alignment. *Front. Artif. Intell.* **8**, 1662984 (2025).

36. Lin, Z. *et al.* Multi-modal contrastive representation learning for entity alignment. *arXiv preprint arXiv:2209.00891* (2022).

37. Snijder, L. L., Smit, Q. T. S. & de Boer, M. H. T. Advancing ontology alignment in the labor market: Combining large language models with domain knowledge. In *Proceedings of the AAAI Symposium Series*, vol. 3, 253–262 (2024).

38. Zhan, R., Yang, X., Wong, D., Chao, L. & Zhang, Y. Prefix text as a yarn: Eliciting non-english alignment in foundation language model. In *Findings of the Association for Computational Linguistics: ACL 2024*, 12131–12145 (2024).

39. Qiang, Z., Taylor, K., Wang, W. & Jiang, J. Oaei-llm: A benchmark dataset for understanding large language model hallucinations in ontology matching. *arXiv preprint arXiv:2409.14038* (2024).

40. Chang, B. *et al.* Dgsvis: Visual analysis of hierarchical snapshots in dynamic graph. *arXiv preprint arXiv:2205.13220* (2022).

41. Gao, S., Li, Y., Shen, Y., Shao, Y. & Chen, L. Etc: Efficient training of temporal graph neural networks over large-scale dynamic graphs. *Proc. VLDB Endow.* **17**, 1060–1072 (2024).

42. Tian, S., Xiong, T. & Shi, L. Streaming dynamic graph neural networks for continuous-time temporal graph modeling. In *2021 IEEE International Conference on Data Mining (ICDM)*, 1361–1366 (2021).

43. Jiang, X. & Pu, Y. Exploring time granularity on temporal graphs for dynamic link prediction in real-world networks. *arXiv preprint arXiv:2311.12255* (2023).

44. Wu, Y., Liu, X., Feng, Y., Wang, Z. & Zhao, D. Jointly learning entity and relation representations for entity alignment. 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)*, 240–249 (2019).

45. Chen, M., Tian, Y., Yang, M. & Zaniolo, C. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 1511–1517 (2017).

46. Wu, Y. *et al.* Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence* (2019).

47. Zhang, S., Tong, H., Xu, J. & Maciejewski, R. Graph convolutional networks: a comprehensive review. *Comput. Soc. Networks* **6**, 1–23 (2019).

48. Zhang, F. *et al.* Cross language entity alignment based on knowledge augmentation of large language model. In *Proceedings of the 2025 2nd International Conference on Computer and Multimedia Technology*, 79–85 (2025).

49. Zhang, Z., Chen, C. L. P., Su, Z. & Zhang, T. Prompts libra: Enhanced image outpainting diffusion model with balanced bimodal guidance. *IEEE Transactions on Circuits Syst. for Video Technol.* (2025).

50. Shen, C. *et al.* Proopf: Benchmarking and improving llms for professional-grade power systems optimization modeling. *arXiv preprint arXiv:2602.03070* (2026).

51. Zhang, Z., Shen, C., Wan, X., Song, J. & Sun, M. Llm-guided safe reinforcement learning for energy system topology reconfiguration. *arXiv preprint arXiv:2603.14018* (2026).

52. Yang, K. *et al.* Llm40fd: Unlocking the potential of llm for anonymous zero-shot fraud detection. *IEEE Transactions on Comput. Soc. Syst.* (2025).

53. Yang, K., Luo, Y., Zhang, Z., Chen, C. L. P. & Zhang, T. Multimodal affect perception with large language model enhancement network. *IEEE Transactions on Affect. Comput.* (2025).

54. Lan, K., Chen, C. L. P., Zhang, Z. & Zhang, T. Contrastive adversarial tuning: Enhancing discriminability and robustness of llms for emotion recognition in conversation. *Pattern Recognit.* **174**, 113025, DOI: 10.1016/j.patcog.2025.113025 (2026).

55. Zhang, R., Li, Z., Yuan, H., Chen, C. L. P. & Zhang, T. P^2-DPO: Grounding hallucination in perceptual processing via calibration direct preference optimization. In *The Fourteenth International Conference on Learning Representations* (2026).

56. Sun, Z., Hu, W. & Li, C. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International Semantic Web Conference*, 628–644 (Springer, 2017).

Data availability

The datasets analyzed during the current study are publicly available in the JAPE repository at https://github.com/nju-websoft/jape. We used the DBP15K benchmark[56], including the DBP15K-ZH-EN, DBP15K-JA-EN, and DBP15K-FR-EN subsets, each of which contains 15,000 gold alignment pairs[17,45].

Acknowledgements

The authors thank Guangdong Power Grid Company Limited for support and discussion during this study.

Funding

This work was supported by the Science and Technology Project of Guangdong Power Grid Co., Ltd., “Research on Key Technologies and Demonstration Application of Marketing Business Knowledge Agents Based on Large Models and Knowledge Graphs” (No. 030000KC24030042).

Author contributions statement

H.Q. conceived the study and drafted the manuscript. H.Q., F.W., B.L., and W.L. designed the framework and organized the study. B.L. and W.L. prepared the technical materials and experimental content. F.W. contributed domain analysis and application requirements. All authors reviewed the manuscript and approved the final version.

Competing interests

The authors declare no competing interests.