Can KBQA Models Predict Their Reasoning Paths? Isomorphism Prediction Task as a Proxy

Zhen Wu and **Ritam Dutt** and **Dhruv Gupta** and **Carolyn Rosé**
Language Technologies Institute, Carnegie Mellon University
{zhenwu, rdutt, dhruvgu2, cprose}@cs.cmu.edu

Abstract

Despite achieving correct answers, we find that existing Knowledge Base Question Answering (KBQA) models struggle to follow the expected reasoning structures. We introduce the task of isomorphism prediction to enhance reasoning fidelity beyond answer generation, with a focus on generalization. We propose a contrastive knowledge co-distillation framework that unifies textual and graphical KBQA paradigms, improving overall isomorphism prediction and model generalization. Furthermore, incorporating isomorphism prediction as an auxiliary task also improves KBQA performance.

1 Introduction and Related Work

The task of question answering over knowledge bases (KBQA) involves reasoning over structured sources of knowledge in the form of knowledge bases (KB) to answer natural language queries. Beyond improved answer accuracy, a key challenge in KBQA lies in understanding how the models perform and ensuring that they faithfully reconstruct the reasoning process. To that end, recent work has leveraged the idea of isomorphisms (Dutt et al., 2023) to characterize the complexity of KBQA questions. Isomorphisms act as a structural proxy for reasoning difficulty by grouping instances that exhibit similar reasoning patterns over the knowledge base. Prior work has explored using isomorphisms as a diagnostic test to investigate the generalization capabilities of KBQA systems. For example, Dutt et al. shows that leveraging gold isomorphisms as inference-time scaffolds improves zero-shot generalization without retraining.

In this work, we introduce the task of **isomorphism prediction** to improve reasoning fidelity in KBQA. Our task formulation is motivated by the observation, that when optimized for answer prediction, KBQA systems are able to generate spurious logical forms that do not conform to the underlying reasoning path but can lead to partial correct answers (Table 3). Furthermore, we observe that predicting the correct isomorphism category is challenging even for large language models (LLMs) (Table 6), highlighting the fact that the task requires models to capture structural dependencies beyond surface-level answer generation. Rather than solely using isomorphisms as a diagnostic tool, we frame them as a learning objective to encourage models to explicitly predict their underlying reasoning structures.

An advantage of this formulation is that it is applicable to both major KBQA paradigms: (i) semantic parsing-based approaches, which translate natural language queries into logical forms (e.g., S-expressions or SPARQL) for execution over the KB (Xie et al., 2022; Ye et al., 2021; Li et al., 2024), and (ii) information retrieval-based approaches, where models directly interact with the knowledge graph to retrieve answers (Das et al., 2022; Dutt et al., 2022; He et al., 2021). Building on this, we also propose a contrastive knowledge co-distillation framework that unifies these two paradigms to enhance isomorphism prediction.

Our experiments show that multitask learning with isomorphism prediction improves both KBQA and isomorphism prediction performance. Additionally, the proposed knowledge co-distillation framework bridges the strengths of both KBQA paradigms and enables better generalization.

2 Preliminaries

2.1 Isomorphism Prediction

We introduce isomorphism prediction to characterize reasoning paths following the definitions in Dutt et al. (2023). Each subgraph $G_i(V_i, E_i)$ represents the logical structure required to answer a question Q_i, where nodes V_i correspond to entities and edges E_i represent relations. Two subgraphs $G_i(V_i, E_i)$ and $G_j(V_j, E_j)$ are considered isomor-

phic if there exists a bijective mapping $\psi : V_i \rightarrow V_j$ between their node sets, such that for all pairs of edges $(m, n) \in E_i$, there exists an corresponding edge in E_j:

$$(m, n) \in E_i \Leftrightarrow (\psi(m), \psi(n)) \in E_j$$

By assigning each subgraph to an isomorphism category C_i, we abstract away entity-specific details and focus purely on the structural reasoning pattern used to derive answers. We present the definitions and examples of each isomorphism type in Table 5.

Models are trained using a multi-class classification objective. We assess isomorphism prediction performance with macro F1-scores.

2.2 KBQA Tasks

Text Model: S-expression Generation Following Xie et al. (2022), the input to the text model consists of a question Q_i and a linearized representation of the subgraph (upper-left in Figure 1). The model generates an S-expression that retrieves the predicted answers when executed on the KB.

Graph Model: Node Classification The graph model operates directly on the subgraph G_i. It assigns probabilities to all nodes in the subgraph, indicating their likelihood of being answers. Training is optimized with binary cross-entropy.

KBQA Evaluation Mechanism We evaluate the aforementioned KBQA tasks with Hits@K, where K is the number of gold answers for a given question. This measures the proportion of correct answers in the top-K predictions.

For the text model, since S-expression generation does not produce ranked outputs, we use beam search to generate N_{beam} S-expression candidates, execute them through the KB, rank the executed answers by frequency, and compute Hits@K likewise. Refer to Section A.3 for detailed equations.

3 Contrastive Knowledge Co-Distillation for Isomorphism Prediction

Our Contrastive Knowledge Co-Distillation framework (Figure 1) consists of two key objectives: isomorphism prediction augmentation and contrastive representation alignment.

3.1 Isomorphism Prediction Augmentation

We employ two parallel encoding pathways. The textual encoder produces a pooled embedding h_t by processing the question along with a linearized subgraph. The graph encoder, implemented as a GNN, directly operates on the structured subgraph and generates a pooled graph-level representation h_g. These representations are concatenated as $h_{concat} = [h_t; , h_g]$ and passed through a classifier optimized via cross-entropy loss:

$$\mathcal{L}_{\text{iso}} = -\sum_i \log P(C_i \mid h_{concat}) \quad (1)$$

3.2 Contrastive Knowledge Co-Distillation

Unlike traditional one-way knowledge distillation, Contrastive Co-Distillation (Yao et al., 2024; Nourbakhsh et al., 2024) (CoD) fosters bidirectional knowledge transfer between text and graph models by contrastive representation learning and stop gradient operation.

As Tian et al. (2022) suggests, contrastive representation learning captures structural information from the teacher's representation space:

$$l_{\text{cl}}(t, s) = -\log \frac{e^{sim(t,s)/\tau}}{\sum_q \mathbb{1}_{[q \neq t]}\, e^{sim(t,q)/\tau}} \quad (2)$$

where t and s are teacher and student representations, q indicates other representations from the training data, $sim(.,.)$ is cosine similarity, τ is temperature.

Based on this, we first define MLP projection heads to map text and graph representations into a shared space: $z_t = \text{MLP}_t(h_t)$ and $z_g = \text{MLP}_g(h_g)$. The CoD loss is computed as:

$$\mathcal{L}_{\text{CoD}} = \frac{1}{2}\sum_i [l_{\text{cl}}(z_i^{\text{text}}, \hat{z}_i^{\text{graph}}) + l_{\text{cl}}(z_i^{\text{graph}}, \hat{z}_i^{\text{text}})] \quad (3)$$

where $\hat{\cdot}$ is the stop gradient operator (Chen and He, 2021) to set the input variable to a constant.

Putting these together, our final objective jointly performs mutual distillation and model optimization end-to-end through a single loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{iso}} + \mathcal{L}_{\text{CoD}} \quad (4)$$

4 Experiments

4.1 Dataset

We employ the WebQuestionsSP (WebQSP) dataset (Yih et al., 2016), a popular benchmark in English for KBQA. Specifically, we use the dataset of Xie et al. (2022) where each question is accompanied with (i) a corresponding subgraph of the Freebase knowledge base (Bollacker et al., 2008)

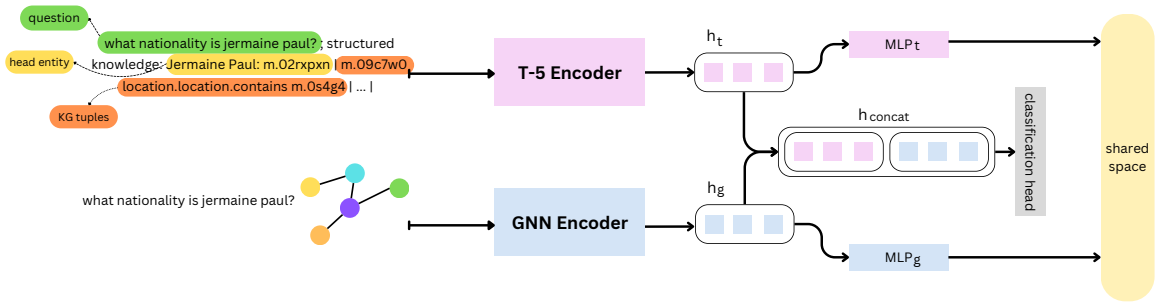


Figure 1: Our contrastive knowledge co-distillation framework. The T5 encoder processes a linearized knowledge subgraph representation, while the GNN directly operates on the KG. Their representations are concatenated for isomorphism classification and projected into a shared space for contrastive co-distillation.

where the answer resides, and (ii) a corresponding logical form in the form of S-expressions or SPARQL-query. Such a design enables us to evaluate the performance of KBQA systems from either a semantic parsing or information retrieval paradigm. Additionally, to investigate different levels of KBQA generalization, we use the approach of Jiang and Usbeck (2022) to obtain a dev or test split with equal proportion of i.i.d., compositional, and zero-shot examples. We present the statistics of our dataset in Table 1 and examples of levels of generalization in Appendix A.2.

Code	Desc.	i.i.d.	Comp	Z.S.	Total
T-0		50.3	0.0	49.7	54.5
T-1		37.3	44.3	18.4	23.5
T-2		17.1	47.1	35.7	5.2
T-3		83.3	6.7	10.0	2.2
T-4		12.8	81.5	5.6	14.5
ALL		40.8	24.9	34.3	100.0

Table 1: Distribution of isomorphisms over the generalization splits (i.i.d., compositional (Comp), zero-shot (Z.S.)) of WebQSP.

4.2 Models

Our primary text model is based on T5 (Raffel et al., 2023), while our graph model is built using Relational Graph Convolutional Network (RGCN) layers (Schlichtkrull et al., 2017). To incorporate question context, we first encode the question using T5 and concatenate this embedding with each node before passing through the GNN. To assess the robustness of or knowledge co-distillation method, for the isomorphism prediction task, we additionally experiment with BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019) to cover a broader range of architectures, and with Relational Graph Attention Networks (RGATs) (Busbridge et al., 2019) as an alternative graph model backbone.

4.3 Experiments

We establish T5 and GNN baselines, each trained separately for isomorphism prediction and their respective KBQA tasks defined in Section 2.2. We evaluate our approach under two settings: 1) CoD framework for isomorphism prediction; 2) Multitask-learning KBQA using isomorphism prediction as an auxiliary task. We report in Section 5 the average performance over multiple seeds.

Moreover, we show that isomorphism prediction is challenging through two diagnostic experiments.

Firstly, we evaluate several widely-used LLMs on the isomorphism prediction task using few-shot prompting. As shown in Table 6, the models struggle to reliably predict isomorphisms.

Further, we analyze whether optimizing for S-expression generation inherently preserves isomorphism structures. Although isomorphism categories can be deterministically derived from S-expressions, models like T5 are trained to optimize answer accuracy rather than faithfully reconstructing reasoning paths. As a result, they may reach correct answers through spurious reasoning rather than the intended structural pattern. Indeed, isomorphism prediction performance drops by 16% overall when inferred post-hoc from generated S-expressions compared with being explicitly learned in our T5 baseline (Table 3). This highlights the importance of directly modeling isomorphisms beyond relying on answer-driven supervision alone.

5 Results and Analysis

Isomorphism Prediction with CoD Overall, CoD outperforms both baselines (ALL in Table

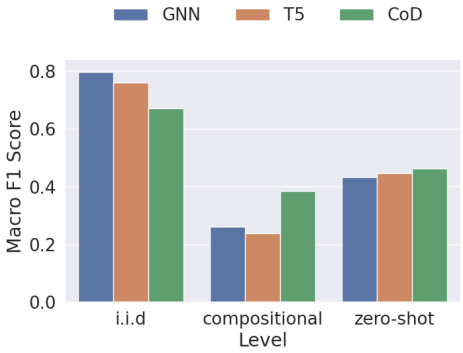


Figure 2: Macro-F1 performance of the models/settings for different generalization levels in WebQSP.

Code	Desc.	T5	GNN	CoD
T-0		**89.1**	88.8	85.9
T-1		60.8	58.8	**68.3**
T-2		38.9	**44.1**	43.5
T-3		45.7	41.1	**59.7**
T-4		50.3	**59.4**	51.8
ALL		57.0	58.4	**61.9**

Table 2: Macro-F1 of different settings (T5, GNN, and CoD) over isomorphism categories in WebQSP. ALL refers to the entire dataset. Best performance in **bold**, second-best underlined.

2). We also observe consistently stronger performance when applying CoD across different model architectures, compared to their non-CoD counterparts (Table 7). We further stratify questions along generalization level and isomorphism category.

For generalization, GNN excels in i.i.d. cases but suffers in generalization, while T5 struggles the most in compositional settings. Although CoD performs lower in the i.i.d. setting, it significantly improves generalization, especially in compositional cases. This suggests that our contrastive knowledge co-distillation enables better adaptation rather than memorizing dataset biases.

Across isomorphism types, all models perform well in T-0 category (single-hop retrieval). As reasoning complexity increases, different model strengths become more evident. T5 performs well in linear chains (T-0, T-1) but struggles with more complex structures, while GNN is better with graph-structured constraints (T-2, T-4) but limited with sequential dependencies (T-3). Notably, CoD significantly improves on T-1 and T-3 and shows

Code	Desc.	T5 (Sexp)	T5 (Iso Pred)
T-0		82.0	**89.1**
T-1		50.3	**60.8**
T-2		37.3	**38.9**
T-3		35.9	**45.7**
T-4		40.1	**50.3**
ALL		40.9	**57.0**

Table 3: F1 performance of T5 on isomorphism prediction when inferred from generated S-expressions versus explicitly predicted as a supervised task.

moderate gains in T-2 and T-4, which indicates that the unification brings together the complementarities of the two models.

Multitask with KBQA and Isomorphism Prediction Table 4 shows that jointly training on KBQA and isomorphism prediction improves performance on both tasks compared to their single-task baselines. Our preliminary result shows that isomorphism prediction provides additional structural supervision, which can help models better capture reasoning patterns beyond answer retrieval.

Model	Task	KBQA	Iso Pred
T5	KBQA only	50.7	-
	Iso Pred only	-	59.0
	Multitask	**52.2**	**61.7**
GNN	KBQA only	54.6	-
	Iso Pred only	-	59.4
	Multitask	**55.3**	**64.0**

Table 4: Comparison of the respective task baselines and the multitask setting using isomorphism prediction as an auxiliary task.

6 Conclusion

We introduce isomorphism prediction task to enhance reasoning fidelity in KBQA. Our contrastive knowledge co-distillation framework improves isomorphism prediction and generalization, particularly in compositional and zero-shot settings. Additionally, isomorphism prediction as an auxiliary task improves KBQA performance, suggesting structural reasoning signals could aid answer generation. Future work can explore broader model architectures and datasets.

7 Limitations

Dataset Diversity Our experiments use WebQSP. Future work could extend evaluations to benchmarks with more diverse KG schemas.

Explicit Isomorphism Learning Future work could explore unsupervised learning to infer reasoning structures without predefined labels.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y. Hammerla. 2019. Relational graph attention networks. *Preprint*, arXiv:1904.05811.

Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758.

Rajarshi Das, Ameya Godbole, Ankita Naik, Elliot Tower, Manzil Zaheer, Hannaneh Hajishirzi, Robin Jia, and Andrew McCallum. 2022. Knowledge base question answering by case-based reasoning over subgraphs. In *International conference on machine learning*, pages 4777–4793. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Ritam Dutt, Kasturi Bhattacharjee, Rashmi Gangadharaiah, Dan Roth, and Carolyn Rose. 2022. Perkgqa: Question answering over personalized knowledge graphs. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 253–268.

Ritam Dutt, Sopan Khosla, Vinayshekhar Bannihatti Kumar, and Rashmi Gangadharaiah. 2023. Grailqa++: A challenging zero-shot benchmark for knowledge base question answering. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–909.

Ritam Dutt, Dongfang Ling, Yu Gu, and Carolyn Penstein Rosé. Leveraging isomorphisms to facilitate zero-shot kbqa generalization.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond iid: three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, pages 3477–3488.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 553–561.

Longquan Jiang and Ricardo Usbeck. 2022. Knowledge graph question answering datasets and their generalizability: Are they enough for future research? *arXiv preprint arXiv:2205.06573*.

Chunhui Li, Yifan Wang, Zhen Wu, Zhen Yu, Fei Zhao, Shujian Huang, and Xinyu Dai. 2024. Multisql: A schema-integrated context-dependent text2sql dataset with diverse sql operations. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13857–13867.

Armineh Nourbakhsh, Zhao Jin, Siddharth Parekh, Sameena Shah, and Carolyn Rose. 2024. AliGATr: Graph-based layout generation for form understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13309–13328, Miami, Florida, USA. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks. *Preprint*, arXiv:1703.06103.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2022. Contrastive representation distillation. *Preprint*, arXiv:1910.10699.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 602–631, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hao-Ren Yao, Luke Breitfeller, Aakanksha Naik, Chunxiao Zhou, and Carolyn Rose. 2024. Distilling multi-scale knowledge for event temporal relation extraction. In *Proceedings of the 33rd ACM*

International Conference on Information and Knowledge Management, CIKM '24, page 2971–2980, New York, NY, USA. Association for Computing Machinery.

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2021. Rng-kbqa: Generation augmented iterative ranking for knowledge base question answering. *arXiv preprint arXiv:2109.08678*.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. 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)*, pages 201–206.

A Appendix

A.1 Isomorphism Examples

See Table 5.

A.2 Examples of Levels of Generalization

We follow Gu et al. (2021) to characterize different levels of generalization for KBQA:

- i.i.d.: These are questions that follow similar logical structures as those seen during training. For instance, the question "Who was the author of Oliver Twist?" and "Who wrote Pride and Prejudice?" share the same underlying reasoning pattern and relation.
- Compositional: These questions involve combinations of relations seen during training but require novel compositions or reasoning steps during inference. For example, both "Who wrote Pride and Prejudice?" and "Who wrote both The Talisman and It?" involve the same "written-by" relation, but the latter demands reasoning over multiple constraints/entities (Talisman, It) and a more complex logical form.
- Zero-shot: These questions require reasoning over relations that are not present in the training set. For example, if training questions focus on the ""written-by" relation, a test question like "Who directed Pride and Prejudice in 2005?" would require handling the unseen "directed-by" relation.

Conceptually, as noted in Dutt et al. (2023), these three levels of generalization are arranged in increasing order of difficulty; with I.I.D. being the least challenging since it operates over templates seen during training, followed by Compositional, which occurs over unseen templates, and then Zero Shot which has unseen schema items.

A.3 Hits@K Computations

Let A_i be the gold answer set for a given question. The graph model ranks all nodes in the subgraph by predicted probabilities. Given a ranked list $\hat{A}_i$, for some $\varepsilon > 0$, Hits@K_{graph} is computed as:

$$Hits@K_{\text{graph}} = \frac{|TopK(\hat{A}_i) \cap A_i|}{K + \varepsilon} \quad (5)$$

Unlike the graph model, the text model does not inherently rank its predictions. To approximate a ranking mechanism, we employ beam search to generate N_{beam} candidate S-expressions $S_{i,j}, j = 1, ..., N_{\text{beam}}$. We then execute these S-expressions through KB to obtain a predicted answer set P_i, and aggregate $P_{i,j}$ by their frequency across all beams. Using this ranked set, for some $\varepsilon > 0$, Hits@K_{text} is computed as:

$$Hits@K_{\text{text}} = \frac{|TopK(Rank(\cup^{N_{\text{beam}}} P_i)) \cap A_i|}{K + \varepsilon} \quad (6)$$

where $Rank(\cup^{N_{\text{beam}}} P_i)$ refers to the aggregated ranking of answer candidates obtained from executing S_i through KB.

A.4 Few-shot LLM on Isomorphism Prediction

We evaluate a couple of widely-used LLMs on the isomorphism prediction task with few-shot prompting, including GPT-3.5-turbo and GPT-4o-mini. As shown in Table 6, these models struggle to reliable predict isomorphisms even with multiple examples per type of isomorphism. We try k-shot prompting with $k = 1, 3$, where we include k examples of each isomorphism type (T-0 to T-4), selected randomly from the training split. The exact prompts used can be found in Appendix A.7.

We experiment with not only the number of few-shot examples provided to the model, but also the technique used to serialize the knowledge graph tuples into a text format as well as the level of detail in the prompt about descriptions of particular isomorphisms. For serializing the knowledge graph tuples of the form (entity$_1$, rel, entity$_2$) we try:

1. **Basic serialization:** where we simply concatenate the knowledge graph tuples using whitespace, for example "entity$_1$ rel entity$_2$".

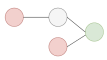
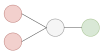
Iso-Type	Illustration	Definition	Example Question	S-expression
T-0		Direct 1-hop connection from constraint to answer	What is the name of money in Brazil?	(JOIN (R location.country.currency_used) m.015fr)
T-1		2-hop linear path	Where does the Queen of Denmark live?	(JOIN (R people.place_lived.location) (JOIN (R people.person.places_lived) m.0g2kv))
T-2		V-pattern with two constraints meeting at a shared node	What was Elie Wiesel's father's name?	(AND (JOIN people.person.gender m.05zppz) (JOIN (R people.person.parents) m.02vsp))
T-3		A chain pattern connecting constraints serially	Where did Joe Namath attend college?	(AND (JOIN common.topic.notable_types m.01y2hnl) (JOIN (R education.education.institution) (JOIN (R people.person.education) m.01p_3k)))
T-4		Y-pattern with merging constraints	Who does Zach Galifianakis play in The Hangover?	(JOIN (R film.performance.character) (AND (JOIN film.performance.film m.0n3xxpd) (JOIN (R film.actor.film) m.02_0d2)))

Table 5: Isomorphism types with their corresponding definitions, example questions, and S-expressions.

2. **Descriptive serialization:** where we concatenate each individual tuple with slightly more description, for example "entity$_1$ is connected to entity$_2$ via relation rel".

We try two levels of isomorphism description detail in our prompt. In the first setting, Prompt 1 (Appendix A.7.1), we provide a brief textual description of each of the isomorphisms' structural characteristics. Whereas in Prompt 2 (Appendix A.7.2), we do not provide any description whatsoever of individual isomorphism categories. The LLMs' final answers are extracted using a regex expression to match the last occurrence of the pattern "T-X", which indicates the model's isomorphism prediction. These predictions are then evaluated using a standard macro F-1 score. These scores, across all experiments, are shown in Table 6. We find that even with few-shot examples, and across all our prompting methods described above, the best performance achieved is a mean macro F-1 of 0.15 by the gpt-3.5-turbo model when given 3 examples per isomorphism class, basic serialized tuples and brief descriptions of isomorphisms' structure.

A.5 Overall Isomorphism Prediction Results

See Table 7.

A.6 Qualitative Analysis on CoD for Dataset Biases Mitigation

We conduct a follow-up analysis by identifying test instances where both T5 and GNN consistently underperform across multiple seeds, but our unified model achieves correct predictions.

Model Configuration	Macro F1
GPT-3.5-turbo	
k=1 (base)	0.09220
k=1 (descriptive tuples)	0.14062
k=1 (descriptive tuples, no iso. desc. in prompt)	0.14118
k=3 (base)	**0.15474**
GPT-4o-mini	
k=1 (base)	0.10530
k=3 (base)	0.10273
k=3 (descriptive tuples)	0.14423
k=3 (descriptive tuples, no iso. desc. in prompt)	0.11832
k=5 (base)	0.07076
k=5 (descriptive tuples)	0.12022
k=5 (descriptive tuples, no iso. desc. in prompt)	0.09122

Table 6: Isomorphism prediction performance of GPT-3.5-turbo and GPT-4o-mini using few-shot prompting. The base configuration refers to when we serialize in a basic manner and provide brief isomorphism descriptions in the prompt.

We observe that the majority of these instances fall into isomorphism types T-1 and T-4, with T-4 being the most challenging category in the WebQSP dataset. For example, in the question "Who does Amy Stiller play in Dodgeball?" (T-4), both T5 and GNN misclassify it as T-1. This indicates that both models likely focus on only one of the two key constraints of "Amy Stiller" and "Dodgeball". In contrast, the unified model correctly predicts T-4, suggesting that it is better able to capture the presence of multiple constraints in such questions.

Moreover, the most frequent setting for these improvements is the "zero-shot" generalization level, where approximately 12% of all zero-shot cases are consistently correct with our unified model. We also observe from Table 2 that our unified CoD

Text model	Graph model	Hybrid (CoD) ↑	Text only	Graph only
T5	RGCN	**0.6190**	0.5700	0.5840
T5	RGAT	**0.6120**	0.5700	0.4966
BERT	RGCN	**0.5999**	0.5835	0.5840
BERT	RGAT	**0.5956**	0.5835	0.4966
GPT-2	RGCN	**0.6022**	0.5614	0.5840
GPT-2	RGAT	**0.6049**	0.5614	0.4966

Table 7: Isomorphism prediction task performance across different model architectures (averaged over multiple seeds). Best performance in **bold**, second-best underlined. Our unified model with knowledge co-distillation (CoD) consistently improves performance across model variants.

framework is particularly helpful for the under-represented isomorphism categories like T-2 and T-3, with approximately 30% improvement over text-only and graph-only baselines on T-3. This demonstrates that our proposed framework can mitigate the biases corresponding to infrequent isomorphism categories.

A.7 Few-shot LLM Prompt

Below are the two versions of prompts we experimented with. Prompt 1 contains brief structural descriptions of each isomorphism category, whereas Prompt 2 simply instructs the model to identify the isomorphism based on the examples provided.

A.7.1 Prompt 1: Structural Descriptions

System prompt: "You are a helpful assistant that identifies isomorphism patterns in knowledge graphs."

User prompt: "Given a question, its entities, and knowledge graph tuples, determine the isomorphism pattern that shows how constraints connect to reach the answer node. In this classification: T-0 means a direct 1-hop connection from constraint to answer, T-1 is a 2-hop linear path, T-2 is a V-pattern with two constraints meeting at a shared node, T-3 is a chain pattern connecting constraints serially, T-4 is a Y-pattern with merging constraints, and T-5+ involve more complex multi-hop patterns. Analyze the structure by tracing the paths from constraints to the answer, counting hops and noting how paths merge or branch. Respond with "Isomorphism: T-X" where X is the pattern number (0-4), output only the final answer. Find some examples below:

{ Question: ...
Entities: (example serialized entities)
Serialized tuples from knowledge graph: (example serialized knowledge graph tuples)
Isomorphism: T-X } (k examples for each type of isomorphism)

###

Question: (target question)
Entities: (target serialized entities)
Serialized tuples from knowledge graph: (target serialized knowledge graph tuples)
Isomorphism:"

A.7.2 Prompt 2: Non-descriptive instructions

System prompt: "You are a helpful assistant that identifies isomorphism patterns in knowledge graphs."

User prompt: "Given a question, its entities, and knowledge graph tuples, determine the isomorphism pattern that shows how constraints connect to reach the answer node. Analyze the structure by tracing the paths from constraints to the answer, counting hops and noting how paths merge or branch. Respond with "Isomorphism: T-X" where X is the pattern number (0-4), output only the final answer. Find some examples below:

{ Question: ...
Entities: (example serialized entities)
Serialized tuples from knowledge graph: (example serialized knowledge graph tuples)
Isomorphism: T-X } (k examples for each type of isomorphism)

###

Question: (target question)
Entities: (target serialized entities)
Serialized tuples from knowledge graph: (target serialized knowledge graph tuples)
Isomorphism:"

A.8 Hyperparameter Settings

On average, our total experiments take around 15 GPU hours.

A.8.1 Experiments on Isomorphism Prediction with CoD

We use the following hyperparameters to obtain results in Table 2.

Model	Batch	Dropout	Others
T5	8	0.2	-
GNN	10	0.2	-
CoD	6	0.3	Weight Decay: 1e-3 Shared Space Dim: 2048
Input Max Length: 512, Patience: 5, LR: 5e-5			

Table 8: Hyperparameters used for experiments in Table 2. Results are averaged over three seeds.

A.8.2 Experiments on Multitask with KBQA and Isomorphism Prediction

We use the following hyperparameters to obtain results in Table 4.

Model	Batch	Dropout	Others
T5 Baseline	10	-	Generation Max Len: 128
GNN Baseline	6	0.2	-
T5 Multitask	10	0.3	Weight Decay: 1e-3 Generation Max Len: 128
GNN Multitask	4	0.2	-
Input Max Length: 512, Patience: 5, LR: 5e-5			

Table 9: Hyperparameters used for KBQA and multitask experiments in Table 4. Results are averaged over three seeds.

A.9 System Specifications

See Table 10.

Component	Specification
GPU	NVIDIA A100 80GB PCIe
CPU	AMD EPYC 7763 (256 vCPUs)
RAM	1TB
CUDA Version	12.6
GPU Memory	80GB

Table 10: Hardware specifications of the computational resources used for experiments.

A.10 Potential Risks and Considerations

Our work builds on WebQSP and Freebase, which may inherit biases from their original data collection. While our focus is on structural reasoning rather than entity-specific biases, these biases could still affect model behavior. Additionally, although we do not train large models from scratch, prompting LLMs, fine-tuning T-5, and training GNN still lead to computational costs, contributing to the environmental footprint.