Similarity-Constrained Reweighting for Complex Query Answering on Knowledge Graphs

Daniel Daza[1,2,7] **Alberto Bernardi**[3] **Luca Costabello**[3] **Christophe Gueret**[3]
Michael Cochez[4,5,6] **Martijn Schut**[7]

[1]Vrije Universiteit Amsterdam
[2]Eindhoven University of Technology
[3]Accenture Labs
[4]ELLIS Institute Finland
[5]Åbo Akademi University, Turku, Finland
[6]Elsevier Discovery Lab, Amsterdam
[7]Translational AI Laboratory, Department of Laboratory Medicine
Amsterdam University Medical Center, Vrije Universiteit Amsterdam

d.dazacruz@vu.nl alberto.bernardi@accenture.com
luca.costabello@accenture.com christophe.gueret@accenture.com
michael.cochez@abo.fi m.c.schut@amsterdamumc.nl

Abstract

Machine learning models for Complex Query Answering (CQA) on knowledge graphs estimate the likelihood of answers that are not reachable via direct traversal. Recent work has proposed extending such queries with *soft entity constraints*, which require the *target variable* to be similar or dissimilar to specified sets of entities. We study a more general formulation where similarity constraints are specified on an arbitrary variable in a query, and introduce SCORE: a computationally efficient and interpretable method for incorporating similarity constraints on an arbitrary variable of a query. SCORE is based on a lightweight and interpretable score adjustment function that only requires tuning two hyperparameters on a validation set. Our experiments show that SCORE effectively incorporates preferences while maintaining strong query answering performance. It significantly improves ranking quality when constraints apply to the target variable and performs effectively in the more general case involving constraints on intermediate variables, and its performance remains robust when applied to several existing methods for CQA. Our implementation is available at https://github.com/dfdazac/score.

1 Introduction

Knowledge graphs (KGs) have emerged as a fundamental data structure for organizing and reasoning about real-world information, encoding millions of entities and their relationships across diverse domains including encyclopedic knowledge [4], biomedical data [15], and scientific literature [17]. Their structured representation enables reasoning tasks that go beyond simple fact retrieval, supporting complex queries that involve multiple logical operations such as conjunction, disjunction, and existential quantification. These capabilities have made KGs relevant in applications such as question answering and recommendation [34], drug discovery [9], and knowledge-based reasoning [16].

However, knowledge graphs are inherently incomplete, missing a substantial fraction of true facts [23]. This incompleteness poses significant challenges for traditional symbolic query processing, which can only retrieve answers that are explicitly present in the graph. To address this limitation, machine

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

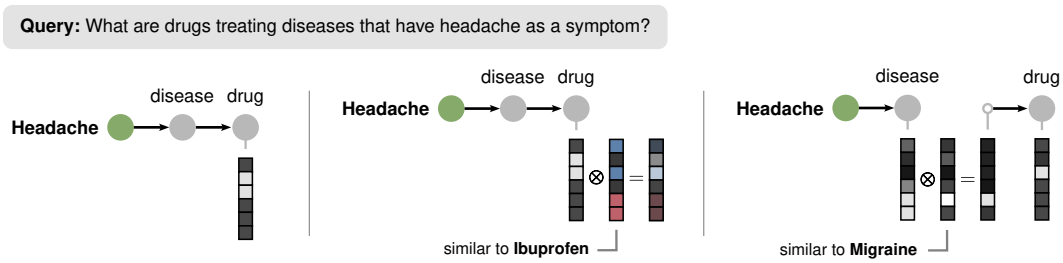


Figure 1: Illustration of three approaches for query answering on knowledge graphs, with vectors indicating entity scores at different steps. **Left**: a traditional query with purely logical constraints. **Middle**: incorporating similarity constraints in NQR [10], which supports only the target variable via uncalibrated score adjustments (shown in red and blue). **Right**: SCORE, which supports similarity constraints on arbitrary variables through normalized similarity scores.

learning approaches for *complex query answering* have been developed [14, 26, 25, 27], which learn to estimate the likelihood of potential answers by reasoning over incomplete knowledge graphs.

Despite these advances, current complex query answering systems are limited to expressing queries through first-order logic, which restricts their ability to capture nuanced preferences and domain-specific requirements. Recent work has begun to address this limitation by introducing *soft entity constraints* [10], which allows specifying that query answers should be similar or dissimilar to given sets of entities. For instance, when querying for "drugs treating diseases that have headache as a symptom," a user might specify preferences such as "drugs similar to Ibuprofen". While this formulation represents a significant step towards more expressive interfaces for querying a KG, it only supports similarity constraints on the final answers being sought, also known as the *target variable*. This limitation limits expressiveness, as constraints can also involve intermediate variables that appear during query processing. For example, one might specify constraints not only on the drug (target variable) but also on the diseases (the intermediate variable), such as "diseases like migraine".

When constraints apply only to target variables, similarity-based score adjustments can be applied as a post-processing step after the query has been evaluated by a base Complex Query Answering (CQA) model. Incorporating constraints on intermediate variables, on the other hand, is a more challenging problem, as they must be *integrated* into the query processing itself. Crucially, this integration depends on operating with *normalized scores*, rather than uncalibrated similarity scores, so that they can be meaningfully combined with other probabilistic components of the query answering model. Such an integration must preserve the logical structure of the query while incorporating similarity constraints at arbitrary positions, which is not supported by existing methods [10].

In this work, we address this fundamental limitation by studying the problem of *Similarity-constrained Complex Query Answering* (SimCQA) in its full generality, where similarity constraints can be specified on **any variable** within a complex query. We then propose SCORE (**S**imilarity-**CO**nstrained **RE**weighting), a computationally efficient method that extends a base CQA model that traverses the query graph by propagating entity score vectors.

The key innovation of SCORE lies in its lightweight and interpretable score adjustment mechanism, which uses pre-trained entity embeddings to compute normalized similarity scores and integrates them into the query processing pipeline, as illustrated in Figure 1 on the right. Unlike approaches that require expensive end-to-end neural optimization, SCORE introduces only two hyperparameters that can be efficiently tuned on a validation set and leaves the base CQA model unchanged. Our contributions are summarized as follows:

- We formalize the problem of similarity-constrained complex query answering with constraints on arbitrary variables, generalizing previous work.
- We introduce SCORE, a lightweight and interpretable method for incorporating similarity constraints at any position in complex queries, requiring minimal computational overhead and hyperparameter tuning.
- We conduct comprehensive experiments on three challenging knowledge graph benchmarks (FB15K-237, NELL-995, and Hetionet), demonstrating that SCORE achieves substantial

improvements in ranking quality and constraint satisfaction that are maintained across different base CQA models.

2 Related Work

Complex query answering. Neural approaches to complex query answering (CQA) have evolved from simple link prediction to multi-hop logical queries [27]. Early knowledge graph embedding methods focused on predicting individual missing edges [22, 5, 31, 19, 29], learning dense representations that capture relational patterns in the graph structure. The field has since progressed to handle complex queries involving multiple entities, relations, and logical operations such as conjunction, disjunction, and existential quantification. Modern *neuro-symbolic* approaches can be broadly categorized based on their query expressiveness: methods supporting smaller subsets of first-order logic [14, 8, 26, 25, 1–3, 35] and those handling more general graph pattern queries [7, 32].

A fundamental limitation shared by all these approaches is their reliance on *hard logical constraints*: queries can only express conditions that are precisely definable in first-order logic. This restriction prevents users from incorporating nuanced preferences or domain-specific similarity requirements that are naturally expressed through soft constraints. Our work addresses this gap by extending neural query answering to support similarity-based preferences while preserving the logical structure and efficiency of existing methods. LLM- and tool-based graph traversal methods [18, 28] address a complementary setting where natural-language questions guide retrieved subgraph exploration, which also requires access to textual descriptions. Our work focuses on graph-structured queries and does not assume access to textual descriptions.

Similarity constraints for complex queries. The first work to extend logical query answering with similarity constraints was introduced by Daza et al. [10], which augmented first-order logic queries with *soft entity constraints* expressing similarity or dissimilarity preferences for query answers. This enriched CQA with preferences that cannot be captured through logical operators alone. However, this prior setting is limited to constraints on the *target variable*, where similarity can be incorporated after query execution as a reranking step. Constraints on intermediate variables create a different computational problem: preference information must enter the query execution process and then be propagated through subsequent projections, intersections, and negations. Uncalibrated post-hoc score shifts are therefore insufficient, because intermediate scores must remain compatible with the fuzzy operators used by the base CQA model. In contrast, SCORE maps similarities into normalized score updates and injects them into intermediate fuzzy vectors, enabling similarity constraints to influence downstream reasoning while preserving the query structure.

3 Preliminaries

Knowledge graphs. We define a knowledge graph as a tuple $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ is the set of binary relations, and $\mathcal{T}$ is the set of triples (h, r, t) with $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$.

Complex queries. Queries over KGs are expressed as first-order logic formulas $q(v_1, \ldots, v_N)$ with free variables $v_1, \ldots, v_N$ ranging over $\mathcal{E}$. Formulas are built from atoms over relations in $\mathcal{R}$ (and their negations), constants denoting specific entities in $\mathcal{E}$, and logical connectives such as conjunction ($\wedge$) and disjunction ($\vee$). For example, consider the question *"What drugs treat diseases that have headache as a symptom?"*. The corresponding query for this question is the following:

$$q(v_1, v_2) = \texttt{SymptomOf}(\texttt{Headache}, v_1) \wedge \texttt{TreatedBy}(v_1, v_2). \tag{1}$$

The free variables in a query can be assigned to specific entities in $\mathcal{E}$. For an assignment $e_1, \ldots, e_N \in \mathcal{E}$, we write $q(e_1, \ldots, e_N)$ for the resulting **grounded statement**. We write $\mathcal{G} \models q(e_1, \ldots, e_N)$ if the set of triples $\mathcal{T}$ in the KG entails this grounded formula.

Continuing the example, suppose $e_1 = \texttt{Migraine}$ and $e_2 = \texttt{Ibuprofen}$. We then obtain the grounded statement:

$$\texttt{SymptomOf}(\texttt{Headache}, \texttt{Migraine}) \wedge \texttt{TreatedBy}(\texttt{Migraine}, \texttt{Ibuprofen}). \tag{2}$$

If both atoms appear in $\mathcal{T}$, then $\mathcal{G} \models q(\texttt{Migraine}, \texttt{Ibuprofen})$.

Answer set. Given a query formula $q(v_1, \ldots, v_N)$, one may choose a free variable v_i as a designated *target variable*. Denoting as $\boldsymbol{v}_{\neg i}$ all free variables except v_i, the answer set $A_i(q)$ is defined as the set of entities that can be assigned to v_i such that the query holds in the graph for some assignment of the remaining variables:

$$A_i(q) = \{e \in \mathcal{E} \mid \exists \boldsymbol{v}_{\neg i} \in \mathcal{E} : \mathcal{G} \models q(\boldsymbol{v}_{\neg i}, v_i = e)\}. \tag{3}$$

Returning to the earlier query in Equation (1), suppose we choose v_2 (the drug) as the target variable. Then the answer set is

$$A_2(q) = \{\, e \in \mathcal{E} \mid \exists v_1 \in \mathcal{E} : \mathcal{G} \models \texttt{SymptomOf}(\texttt{Headache}, v_1) \wedge \texttt{TreatedBy}(v_1, e) \,\}. \tag{4}$$

These definitions form the basis of prior work on complex query answering (CQA), where the goal is to predict answer sets correctly when the graph is incomplete (requiring accurate estimation of the likelihood of entailments $\mathcal{G} \models q(a_1, \ldots, a_N)$) [14, 8, 26, 25]. They are also adopted by Daza et al. [10] to introduce similarity constraints over a single target variable.

4 Similarity-constrained Complex Query Answering

We now consider the more general problem where similarity constraints are specified over an arbitrary variable in a query.

Similarity constraints. A similarity constraint is intended to capture notions such as "diseases like migraine". Formally, a similarity constraint is a boolean predicate s such that for an entity $e \in \mathcal{E}$, $s(e)$ is true if e meets the similarity constraint, and false otherwise.

Constrained answer set. Similarity constraints allow modifying the answer set (Equation (3)) by additionally restricting certain variables to meet the constraint. Given a similarity constraint s applied to variable v_j in a query $q(v_1, \ldots, v_N)$, we denote the *constrained* answer set as follows:

$$\hat{A}_i(q, s, j) = \{e \in \mathcal{E} \mid \exists \boldsymbol{v}_{\neg i} : (\mathcal{G} \models q(\boldsymbol{v}_{\neg i}, v_i = e)) \wedge s(v_j)\}. \tag{5}$$

This can be read as the set of entities e that may serve as answers for v_i, for which there exists an assignment of the remaining variables such that the query holds in the KG, *and* the assignment to v_j satisfies the similarity constraint s.

Because similarity constraints may vary across queries and domain context, s is not fully observed. Instead, we assume access to a small set of k (in the order of tens) labeled examples in a *preference set*:

$$P_s = \{(e_1, s(e_1)), \ldots, (e_k, s(e_k))\}. \tag{6}$$

Suppose the similarity constraint expresses the notion of *"diseases like migraine"*. Then an example preference set is $P_s = \{(\texttt{Migraine}, \text{true}), (\texttt{ClusterHeadache}, \text{true}), (\texttt{Asthma}, \text{false})\}$. Together, these definitions provide the foundation for the problem we study in this work.

> **Similarity-constrained Complex Query Answering.** Given a knowledge graph $\mathcal{G}$, a query $q(v_1, \ldots, v_N)$ with target variable v_i, and a similarity constraint s applied to variable v_j, the task of Similarity-constrained Complex Query Answering (SimCQA) consists of computing the constrained answer set $\hat{A}_i(q, s, j)$ (Equation (5)), where the similarity constraint s is estimated from a preference set P_s (Equation (6)).

This formulation subsumes prior work. If we set $v_i = v_j$, we obtain the setting of Daza et al. [10] where similarity constraints are applied to a single target variable. If, in addition, we define a similarity function $s(e) = \text{true}$ for all $e \in \mathcal{E}$, we recover standard CQA with purely logical queries.

5 Similarity-Constrained Reweighting

As described in Section 2, several neuro-symbolic methods exist for CQA. A common approach consists of computing one **fuzzy vector** of scores $\mathbf{v}_i \in [0, 1]^{|\mathcal{E}|}$ per variable in a query $q(v_1, \ldots, v_N)$.

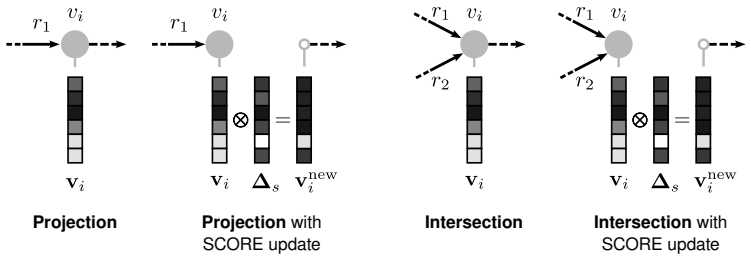


Figure 2: Existing CQA methods compute fuzzy vectors via *projection* and *intersection* operators. We illustrate how we integrate the SCORE update (Equation (9)), by applying it after these operations have been computed by the base CQA model, with $\boldsymbol{\Delta}_s$ indicating similarity scores.

Each entry in the vector indicates the likelihood of each entity in $\mathcal{E}$ to be assigned to v_i. Some examples of methods following this approach are CQD [1, 2], QTO [3], GNN-QE [35], and Ultra-Query [11]. These methods rely on *projection* and *intersection* operators to model the logical query with fuzzy vectors, as illustrated in Figure 2. Once a CQA method has processed a query, the fuzzy vector for the target variable is used as the score for each entity in the KG to answer the query. We describe the fundamental operations used by CQA methods to compute fuzzy vectors for a query in Section A.1. Here, we focus on how to incorporate similarity constraints into this computation.

Similarity-constrained Reweighting (SCORE). The key difference between CQA and SimCQA is the inclusion of a binary predicate in the constrained answer set that restricts one of the variables v_j to meet the similarity constraint s. In principle, this suggests modifying the fuzzy vector $\mathbf{v}_j$ computed by an underlying CQA model by increasing or decreasing the score for each entity depending on whether $s(e)$ is true or false. There are, however, two challenges: (i) the true similarity constraint s is unobserved, and instead it must be approximated from the preference set P_s, and (ii) any adjustments made to a fuzzy vector will have an effect in the final answers to the query as the vector is propagated by the underlying CQA method throughout the rest of the query.

We address these challenges with Similarity-constrained Reweighting (SCORE). Intuitively, given a preference set P_s and a fuzzy vector $\mathbf{v}_j$ computed by a base CQA model, SCORE promotes entities that resemble positively labeled, and demotes those resembling negative ones. As we illustrate in Figure 2, we apply updates to a fuzzy vector after the base CQA model has applied a projection or intersection operator. This *reweighted* vector is then propagated through the rest of the query using the base CQA model.

To approximate the true similarity constraint, SCORE relies on entity embeddings obtained from the base CQA model. We employ these embeddings to compute the cosine similarity $\text{sim}(e_1, e_2) \in [-1, 1]$ between pairs of entities $e_1, e_2 \in \mathcal{E}$. In order to combine this value with fuzzy vectors whose elements lie in the interval $[0, 1]$, we compute a *normalized similarity score*, defined as follows:

$$\overline{\text{sim}}(e_1, e_2) = \tfrac{1}{2}\big(\text{sim}(e_1, e_2) + 1\big). \tag{7}$$

This transformation captures the notion that $\overline{\text{sim}}(e_1, e_2) \in [0, 1]$ tends to 0 if the entities are not similar, and to 1 if they are similar.

SCORE computes updates to a fuzzy vector in *logit space*. For a probability p, the logit is defined as $\text{logit}(p) = \log(\frac{p}{1-p})$. Logits provide an additive scale for combining independent sources of evidence, which stands in contrast with multiplicative updates that can vanish quickly when values are in the interval $[0, 1]$. In preliminary experiments, a multiplicative update in probability space indeed performed worse, as scores vanished when several similarity terms were close to zero, especially when constraints were applied before further query operations.

We define P_s^+ as the set of entities e in P_s for which $s(e_i) =$ true, and similarly P_s^- for the case where $s(e_i) =$ false. The SCORE update consists of two steps:

1. For each entity $e \in \mathcal{E}$, compute the mean logit similarity:

$$\Delta(P_s^{\odot}, e) = \frac{1}{|P_s^{\odot}|} \sum_{e_i \in P_s^{\odot}} \text{logit}(\overline{\text{sim}}(e_i, e)), \tag{8}$$

where $P_s^{\odot} \in \{P_s^+, P_s^-\}$.

2. Reweigh the score of each entity based on the mean logit similarities in logit space, and transform back to the interval $[0, 1]$ with the sigmoid function:

$$\mathbf{v}_j^{\text{new}}[e] = \sigma\big(\text{logit}(\mathbf{v}_j[e]) + w_p\Delta(P_s^+, e) - w_n\Delta(P_s^-, e)\big). \tag{9}$$

SCORE thus only contains two hyperparameters $w_p, w_n \in \mathbb{R}$ which can be tuned to balance the contributions from preferred and non-preferred sets of entities, using a small validation set.

5.1 Theoretical properties

We now highlight three important properties of SCORE. We refer to Section A.2 for proofs of the propositions and details on the connection with Bayesian inference.

Proposition 1 (Monotonicity of SCORE updates)**.** *For any two entities $e_1, e_2 \in \mathcal{E}$, suppose their preference contributions are equal:*

$$w_p\Delta(P_s^+, e_1) - w_n\Delta(P_s^-, e_1) = \; w_p\Delta(P_s^+, e_2) - w_n\Delta(P_s^-, e_2). \tag{10}$$

Then the relative ordering of their fuzzy scores is preserved under the SCORE update:

$$\mathbf{v}_j[e_1] < \mathbf{v}_j[e_2] \implies \mathbf{v}_j^{new}[e_1] < \mathbf{v}_j^{new}[e_2].$$

The monotonicity of SCORE contributes to interpretability: whenever the score of entity is promoted or demoted, the exact source of this shift can be traced back to $\Delta(P_s^+, e)$ and $\Delta(P_s^-, e)$. This makes it transparent whether the change is due to evidence from positively or negatively labeled examples, providing a direct explanation of how similarity constraints affect the final scores.

Proposition 2 (Linear complexity of SCORE)**.** *The computational complexity of applying SCORE is $O(|\mathcal{E}|)$, i.e. linear with respect to the number of entities.*

Connection with posterior log-odds. The SCORE update reflects the log-odds update rule in Bayesian inference. This provides a principled justification for our formulation, in which preferences act as additional likelihood terms that adjust the scores of the base CQA model in a coherent way.

6 Experiments

In our experiments, we follow the ranking-based evaluation which has been employed in works on link prediction [17] and complex query answering [27]. In this setting, the set of triples $\mathcal{T}$ is split into disjoint sets $\mathcal{T}^{\text{train}}$, $\mathcal{T}^{\text{valid}}$, and $\mathcal{T}^{\text{test}}$, and the answer set $A_i(q)$ contains answers that are only reachable by traversing edges in $\mathcal{T}^{\text{test}}$. The goal is to determine the rank assigned by a model to the entities in the answer set $A_i(q)$ when given access to the edges in $\mathcal{T}^{\text{train}} \cup \mathcal{T}^{\text{valid}}$. Similar to prior work [10], in SimCQA the goal is the same, with the addition that entities in the *constrained* answer set $\hat{A}_i(q, s, j)$ should be ranked higher than those not in it.

Datasets. We conduct experiments on knowledge graphs of different domains and scales: Hetionet [15] is a biomedical KG covering genes, diseases, and drugs, among other biomedical entities. FB15k-237 [4, 30] and NELL995 [6] are encyclopedic KGs that contain general facts about people, organizations, and locations. We adopt the complex queries of Ren and Leskovec [25], which cover a wide variety of 14 types of complex queries involving conjunctions, disjunctions, and negations.

In Daza et al. [10], the authors build a benchmark of preference sets from clusters of query answers derived from textual embeddings, which in a human validation subset results in 79% agreement. This agrees with findings that textual embedding similarity correlates with human judgments of similarity [24, 12, 21]. We extend instances in this benchmark with constraints on arbitrary variables. We present additional details of the datasets in Section A.3.

Baselines. We consider three competitive methods as base CQA models: QTO [3], GNN-QE [35], and UltraQuery [11]. Given a base CQA model, we apply SCORE to compute score updates for the SimCQA problem, in two settings where (1) similarity constraints are applied to the target variable (**target-variable** SimCQA), and (2) to any variable (**general** SimCQA).

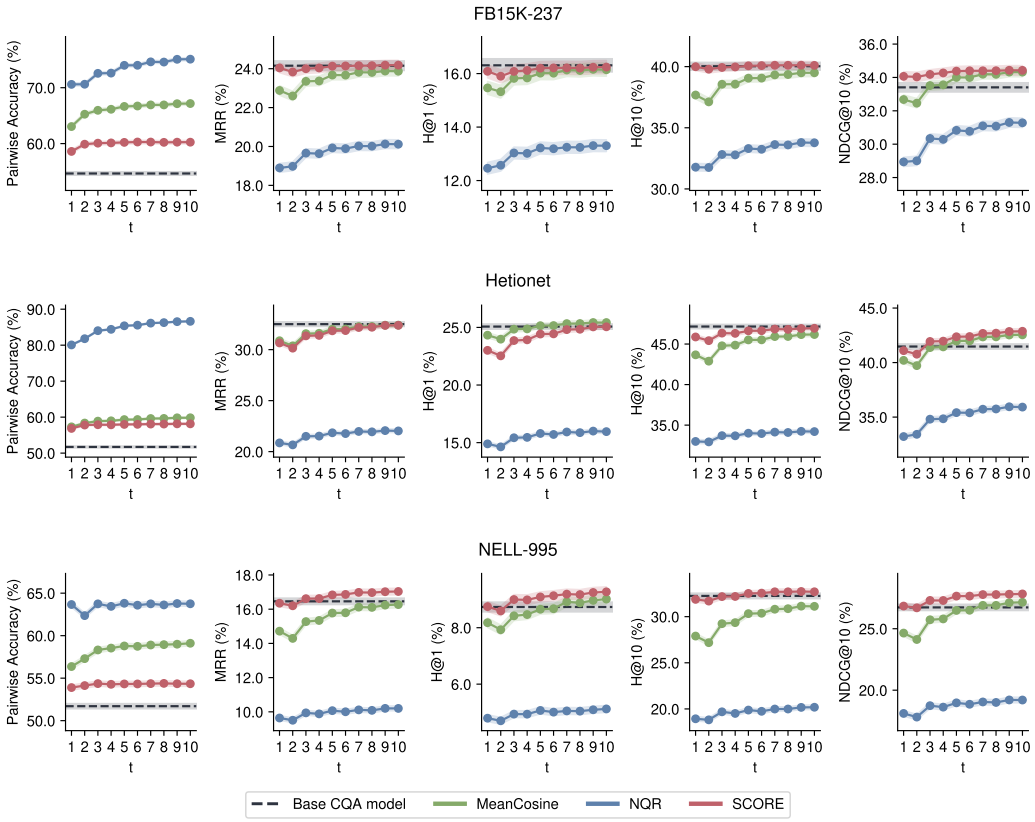


Figure 3: SimCQA results for similarity constraints on the **target** variable. Shaded areas indicate a 95% confidence interval.

We compare the performance of SCORE against NQR [10], a neural reranker that, unlike SCORE, requires training with hundreds of thousands of instances of queries and their corresponding constrained answer sets; and MeanCosine: a simple baseline based on raw cosine similarity values, which modifies the fuzzy vector by adding/subtracting the mean cosine similarity with respect to positive/negative examples [10]. For general SimCQA, we adapt MeanCosine by applying its update to the intermediate fuzzy vector and mapping the result to $[0, 1]$ with a sigmoid before query execution continues. To ensure a fair comparison, we run an extensive hyperparameter grid search for SCORE, NQR, and MeanCosine separately for each dataset, selecting the best values on the validation set. More experimental details are available in Section A.5.

Metrics. We evaluate each method in an **interactive setting** [10], where preference sets P_s of increasing size $t = 1, \ldots, 10$ are provided and performance is measured at each step. We partition the answer set $A_i(q)$ into answers in the constrained answer set $\hat{A}_i(q, s, j)$, which we denote as $A_i(q)^+$, and those not in it, denoted as $A_i(q)^-$. We report the following (see Section A.4 for details):

- **MRR** and **Hits@k (H@k)** measure whether a method preserves the ranking quality of the base CQA model, i.e., how well entities in the unconstrained answer set $A_i(q)$ are ranked.
- **Pairwise Accuracy (PA)** isolates similarity-constraint satisfaction, assessing if entities in the constrained set $\hat{A}_i(q, s, j)$ are ranked above those in $A_i(q)^-$.
- **NDCG@k** combines both aspects by assigning graded relevance (0 to non-answers, 1 to $A_i(q)^-$, and 2 to $A_i(q)^+$), rewarding a balance of global quality with constraint satisfaction.

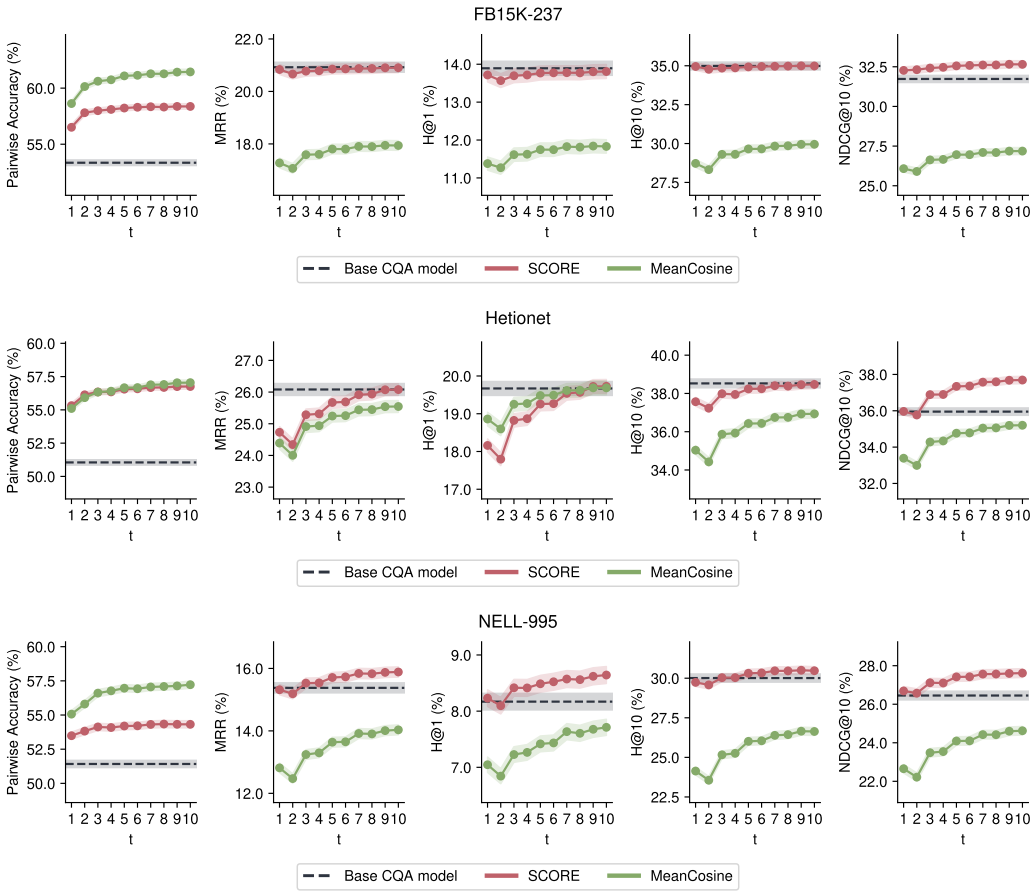


Figure 4: SimCQA results for similarity constraints on the **target and intermediate** variables, comparing SCORE with the adapted MeanCosine baseline. Shaded areas indicate a 95% confidence interval.

6.1 Results

Target-variable SimCQA. We present results for target-variable SimCQA in Figure 3, averaged across all queries and shown with 95% confidence intervals, using QTO [3] as the **base CQA model**.

SCORE achieves the highest NDCG@10 across all datasets, indicating that its resulting ranking more closely resembles the optimal ranking where preferred answers are ranked above non-preferred ones, which are in turn ranked higher than entities not answering a query. Its MRR and H@10 remain close to the base CQA model, which is desirable: SCORE improves preference satisfaction without disrupting the base model's answer ranking. SCORE performs particularly well on small preference sets, indicating that it can quickly adapt to a few labeled examples.

The MeanCosine baseline shows a competitive NDCG@10, but for small preference sets, SCORE generally exhibits larger performance. This is desirable as it indicates that SCORE requires fewer entities in a preference set to capture preferences while retaining global ranking quality. While MeanCosine results in higher pairwise accuracy, we observe that it comes at the cost of degradation in ranking quality, based on the lower values of MRR and H@10. H@1 and H@3 results (the latter included in Section A.6) confirm that SCORE also preserves top-rank answer quality.

The neural NQR model achieves high pairwise accuracy, indicating strong preference satisfaction, but it does so in a more aggressive way that harms global ranking quality. While NQR is designed to balance the two goals [10], these results suggest that NQR can over-optimize local similarity signals at the expense of global ranking fidelity, whereas SCORE preserves ranking structure more reliably.

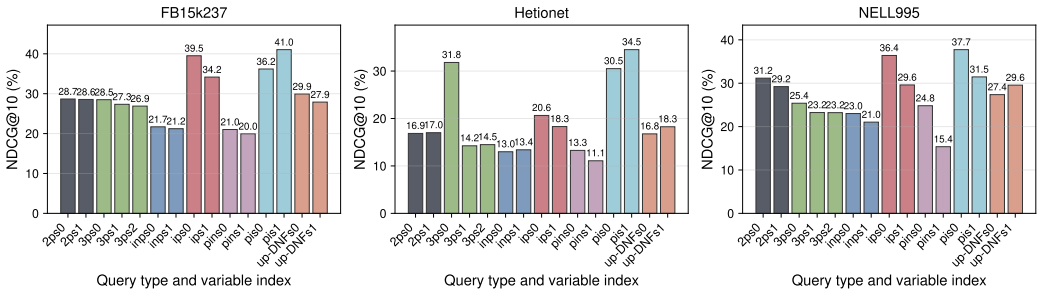


Figure 5: NDCG@10 per query type and variable position on different datasets. Indices indicate where the positions of variables in which similarity constraints are applied in the query.

General SimCQA. In this setting, preferences may be applied before the target variable is reached, so score updates must remain compatible with the base CQA model as query execution continues. As shown in Figure 4, SCORE steadily improves NDCG@10 as more preference examples are provided, indicating that preferred answers move up in the ranking while global answer quality is preserved. The adapted MeanCosine baseline captures pairwise preferences more aggressively, but substantially degrades global ranking quality: on FB15k-237 at $t = 10$, it raises pairwise accuracy from 53.3 to 61.4 but lowers MRR from 20.9 to 17.9, whereas SCORE reaches 58.4 while preserving the MRR of 20.9. This indicates that bounding unnormalized updates with a sigmoid is insufficient for reliable propagation through fuzzy operators. For reference, an oracle that places preferred answers above non-preferred ones while preserving the base ranking otherwise would reach 100% pairwise accuracy at the base MRR. SCORE closes 10.9% of this gap without any training.

Robustness across base CQA models. We present further results on both target-variable and general SimCQA when replacing the base CQA model with GNN-QE [35] and UltraQuery [11] in Section A.6 (Figures 9 and 10). As in our previous results, SCORE improves preference satisfaction and ranking metrics with growing preference sets. The gains are larger with GNN-QE and UltraQuery than with QTO, indicating that these models benefit more from SCORE.

Per-query type results. We present detailed results for each of the 14 query types in Section A.6. SCORE performs best for the large majority of query structures; the few cases where MeanCosine is slightly better do not form a consistent trend across datasets.

Noisy preferences and embedding quality. In order to study the robustness of SCORE to noisy preference sets, we perform an experiment on FB15k-237 where we flip the labels of a fraction of the entities in each preference set. We observe that SCORE remains above the base CQA model in pairwise accuracy with up to 40% flipped labels, while MRR is essentially unchanged. We provide additional details in Section A.7. Similarly, to study how SCORE depends on the quality of entity embeddings, we measure how well the embedding space separates preferred from non-preferred entities. We observe that this alignment is weakly but consistently associated with gains in pairwise accuracy, while it is not associated with MRR. We provide additional details in Section A.8.

Multiple constraints. In order to understand whether SCORE extends to multiple simultaneous constraints, we perform an experiment on FB15k-237 where similarity constraints are applied to two variables of 2p and 3p queries at the same time. We observe that SCORE improves pairwise accuracy while MRR remains close to the base CQA model. We provide additional details in Section A.9.

Runtime overhead. Table 2 reports average query execution times (in ms) with and without SCORE. The additional cost is modest (from 1.24 to 8.45 ms per query) and does not grow superlinearly with dataset size, confirming that SCORE preserves the efficiency of existing CQA methods.

Table 2: Average runtime per query (ms) with and without SCORE.

Dataset	Base	+SCORE	Δ
FB15k237	2.96	4.20	+1.24
Hetionet	25.65	28.68	+3.03
NELL-995	75.51	83.96	+8.45

Table 1: Example 2-hop query from Hetionet, showing the top-scored entities for both the intermediate (drug) and target (side effect) variables before and after applying similarity constraints with SCORE. ▲ indicates a promoted entity, ▼ a demoted one.

$q(v_1, v_2) = \texttt{TreatedBy}(\texttt{Migraine}, v_1) \wedge \texttt{SideEffect}(v_1, v_2)$
$P_s^+ = \{\text{Diclofenac, Naproxen, Prednisone}\}$, $P_s^- = \{\text{Rizatriptan, Orphenadrine, Ergotamine}\}$.

Initial Top-5		**Top-5 after SCORE**	
v_1	v_2	v_1	v_2
Sumatriptan	Hemiplegia transient	▲ Diclofenac	▲ Hypersensitivity
Ergotamine	Cluster headache	▲ Naproxen	▲ Dermatitis
Frovatriptan	Arteriospasm coronary	▲ Prednisone	▲ Pain
Antipyrine	Temporal arteritis	▼ Sumatriptan	▲ Headache
Naratriptan	Hypertensive episode	▼ Frovatriptan	▲ Asthma
	NDCG@10: 28.4		NDCG@10: 36.5

Influence of variable position. Figure 5 reports results by types of queries containing more than one variable, and by the variable position on which similarity constraints are applied. We specify an index that indicates the topological order of variables in the query (see Figure 6 in Section A.3). For example, in a 2-hop query, `2ps0` applies similarity constraints to the intermediate variable, while `2ps1` applies them to the target variable. In most cases, constraints applied *earlier* in the query (i.e., on intermediate variables) yield slightly higher performance than when applied to later variables. This suggests that preference information is more effectively propagated when introduced at earlier stages of the query, reinforcing the value of supporting constraints on arbitrary variables beyond the target.

Qualitative example. Table 1 illustrates how SCORE applies similarity constraints to an intermediate variable. The base CQA model (QTO in this example) retrieves drugs from triptan and ergot classes, together with unrelated side effects, reaching an NDCG@10 of 28.4. With preferences favoring anti-inflammatory drugs and steroids, SCORE shifts the ranking to those compounds and their associated side effects, improving NDCG@10 to 36.5.

Limitations and future work. In this work, similarity constraints are specified through example-based preference sets, and SCORE approximates them with cosine similarities between entity embeddings. Its effectiveness thus depends on how well the embedding space reflects the intended preference and on the quality of the examples: SCORE tolerates moderate label noise, but predominantly misleading feedback reduces preference satisfaction. A promising direction for future work is to support preferences expressed directly in natural language, including hybrid pipelines where LLM- or tool-guided traversal proposes candidates before SCORE reweights full entity score vectors. In addition, it remains interesting to investigate how to incorporate similarity constraints into methods that embed an entire query into a vector, also known as Skolemization methods [20].

7 Conclusion

We introduce and formalize the general problem of Similarity-constrained Complex Query Answering (SimCQA), extending logical queries with similarity constraints on arbitrary variables. To address this problem, we propose Similarity-Constrained Reweighting (SCORE), a computationally efficient and interpretable reweighting layer for base CQA models that propagate entity score vectors. Experiments across multiple knowledge graphs and base CQA models show that SCORE progressively incorporates similarity constraints as preference sets grow, improving preference satisfaction while preserving or improving query answering performance. We further find that constraints applied earlier in the query often yield stronger gains by allowing preference information to propagate more effectively. These results highlight SCORE as a practical and general approach for specifying constraints beyond first-order logic in CQA.

Acknowledgments

Daniel Daza was funded by a gift from Accenture LLP. The work by Michael Cochez has received funding from the European Union under the CEREBRA-AI project (Grant Agreement No. 101299046). Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HaDEA). Neither the European Union nor HaDEA can be held responsible for them. Work on this publication is in part based upon work from COST Action CA23147 GOBLIN - Global Network on Large-Scale, Cross-domain and Multilingual Open Knowledge Graphs, and COST Action CA24121 - Knowledge Graphs in the Era of Large Language Models (KGELL), both supported by COST (European Cooperation in Science and Technology, https://www.cost.eu)

References

[1] Erik Arakelyan, Daniel Daza, Pasquale Minervini, and Michael Cochez. Complex query answering with neural link predictors. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Mos9F9kDwkz.

[2] Erik Arakelyan, Pasquale Minervini, Daniel Daza, Michael Cochez, and Isabelle Augenstein. Adapting neural link predictors for data-efficient complex query answering. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[3] Yushi Bai, Xin Lv, Juanzi Li, and Lei Hou. Answering complex logical queries on knowledge graphs via query computation tree optimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 1472–1491. PMLR, 2023. URL https://proceedings.mlr.press/v202/bai23b.html.

[4] Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In Jason Tsong-Li Wang, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250. ACM, 2008. doi: 10.1145/1376616.1376746.

[5] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

[6] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr, and Tom M. Mitchell. Toward an Architecture for Never-Ending Language Learning. In Maria Fox and David Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, pages 1306–1313. AAAI Press, 2010. doi: 10.1609/AAAI.V24I1.7519. URL https://doi.org/10.1609/aaai.v24i1.7519.

[7] Tamara Cucumides, Daniel Daza, Pablo Barcelo, Michael Cochez, Floris Geerts, Juan L Reutter, and Miguel Romero Orth. Unravl: A neuro-symbolic framework for answering graph pattern queries in knowledge graphs. In *The Third Learning on Graphs Conference*, 2024. URL https://openreview.net/forum?id=183XrFqaHN.

[8] Daniel Daza and Michael Cochez. Message passing query embedding. In *ICML Workshop - Graph Representation Learning and Beyond*, 2020. URL https://arxiv.org/abs/2002.02406.

[9] Daniel Daza, Dimitrios Alivanistos, Payal Mitra, Thom Pijnenburg, Michael Cochez, and Paul Groth. Bioblp: a modular framework for learning on multimodal biomedical knowledge

graphs. *J. Biomed. Semant.*, 14(1):20, 2023. doi: 10.1186/S13326-023-00301-Y. URL https://doi.org/10.1186/s13326-023-00301-y.

[10] Daniel Daza, Alberto Bernardi, Luca Costabello, Christophe Gueret, Masoud Mansoury, Michael Cochez, and Martijn Schut. Interactive query answering on knowledge graphs with soft entity constraints. *arXiv preprint arXiv:2508.13663*, 2025.

[11] Michael Galkin, Jincheng Zhou, Bruno Ribeiro, Jian Tang, and Zhaocheng Zhu. A Foundation Model for Zero-shot Logical Query Reasoning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/616521c3cf15f9f7018565c427d40e3b-Abstract-Conference.html.

[12] Gabriel Grand, Idan A. Blank, Francisco Pereira, and Evelina Fedorenko. Semantic projection recovers rich human knowledge of multiple object features from word embeddings. *Nature Human Behaviour*, 6(7):975–987, Jul 2022. doi: 10.1038/s41562-022-01316-8.

[13] Petr Hájek. *Metamathematics of Fuzzy Logic*, volume 4 of *Trends in Logic*. Kluwer, 1998. ISBN 978-1-4020-0370-7. doi: 10.1007/978-94-011-5300-3. URL https://doi.org/10.1007/978-94-011-5300-3.

[14] William L. Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. Embedding logical queries on knowledge graphs. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 2030–2041, Red Hook, NY, USA, 2018. Curran Associates Inc.

[15] Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio E Baranzini. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife*, 6:e26726, September 2017. ISSN 2050-084X. doi: 10.7554/eLife.26726. URL https://doi.org/10.7554/eLife.26726. Publisher: eLife Sciences Publications, Ltd.

[16] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. *Knowledge Graphs*. Number 22 in Synthesis Lectures on Data, Semantics, and Knowledge. Springer, 2021. ISBN 978-3-031-00790-3. doi: 10.2200/S01125ED1V01Y202109DSK022. URL https://kgbook.org/.

[17] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Networks Learn. Syst.*, 33(2):494–514, 2022. doi: 10.1109/TNNLS.2021.3070843. URL https://doi.org/10.1109/TNNLS.2021.3070843.

[18] Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. StructGPT: A general framework for large language model to reason over structured data. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.574. URL https://aclanthology.org/2023.emnlp-main.574/.

[19] Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2869–2878. PMLR, 2018. URL http://proceedings.mlr.press/v80/lacroix18a.html.

[20] Yuyin Lu, Hegang Chen, Shanrui Xie, Yanghui Rao, Haoran Xie, Fu Lee Wang, and Qing Li. Exploring the paradigm shift from grounding to skolemization for complex query answering on knowledge graphs. *arXiv preprint arXiv:2509.10837*, 2025.

[21] Danny Merkx, Stefan Frank, and Mirjam Ernestus. Seeing the advantage: visually grounding word embeddings to better capture human semantic knowledge. In Emmanuele Chersoni, Nora Hollenstein, Cassandra Jacobs, Yohei Oseki, Laurent Prévot, and Enrico Santus, editors, *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 1–11, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.cmcl-1.1. URL https://aclanthology.org/2022.cmcl-1.1/.

[22] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 809–816, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

[23] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proc. IEEE*, 104(1):11–33, 2016. doi: 10.1109/JPROC.2015.2483592. URL https://doi.org/10.1109/JPROC.2015.2483592.

[24] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1410. URL https://doi.org/10.18653/v1/D19-1410.

[25] Hongyu Ren and Jure Leskovec. Beta Embeddings for Multi-Hop Logical Reasoning in Knowledge Graphs. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e43739bba7cdb577e9e3e4e42447f5a5-Abstract.html.

[26] Hongyu Ren, Weihua Hu, and Jure Leskovec. Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=BJgr4kSFDS.

[27] Hongyu Ren, Mikhail Galkin, Zhaocheng Zhu, Jure Leskovec, and Michael Cochez. Neural graph reasoning: A survey on complex logical query answering. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=xG8un9ZbqT.

[28] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=nnVO1PvbTv.

[29] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HkgEQnRqYQ.

[30] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In Alexandre Allauzen, Edward Grefenstette, Karl Moritz Hermann, Hugo Larochelle, and Scott Wen-tau Yih, editors, *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, CVSC 2015, Beijing, China, July 26-31, 2015*, pages 57–66. Association for Computational Linguistics, 2015. doi: 10.18653/V1/W15-4007. URL https://doi.org/10.18653/v1/W15-4007.

[31] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and*

Conference Proceedings, pages 2071–2080. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/trouillon16.html.

[32] Hang Yin, Zihao Wang, and Yangqiu Song. Rethinking Complex Queries on Knowledge Graphs with Neural Link Predictors. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=1BmveEMNbG.

[33] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965. ISSN 0019-9958. doi: https://doi.org/10.1016/S0019-9958(65)90241-X. URL https://www.sciencedirect.com/science/article/pii/S001999586590241X.

[34] Sijin Zhou, Xinyi Dai, Haokun Chen, Weinan Zhang, Kan Ren, Ruiming Tang, Xiuqiang He, and Yong Yu. Interactive recommender system via knowledge graph-enhanced reinforcement learning. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 179–188, 2020.

[35] Zhaocheng Zhu, Mikhail Galkin, Zuobai Zhang, and Jian Tang. Neural-Symbolic Models for Logical Queries on Knowledge Graphs. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 27454–27478. PMLR, 2022. URL https://proceedings.mlr.press/v162/zhu22c.html.

A Appendix

A.1 Neuro-symbolic methods for CQA

This section provides an overview of the logical operators in the logical queries we consider in this work, as well as the relaxed operators used to model logical operators in existing methods for neuro-symbolic CQA that can be used as base methods for SimCQA.

Logical operators. We consider logical queries over a KG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ specified by a formula $q(v_1, \ldots, v_N)$ in first-order logic, written in disjunctive normal form (DNF), i.e., as a disjunction ($\vee$) of conjunctions ($\wedge$) of *atoms*:

$$q(v_1, \ldots, v_N) = (c_1^1 \wedge \cdots \wedge c_{m_1}^1) \vee \cdots \vee (c_1^n \wedge \cdots \wedge c_{m_n}^n), \tag{11}$$

where each c_j^i denotes a predicate applied to two variables, or a variable and a known entity in $\mathcal{E}$, while optionally including negations:

$$c_j^i = \begin{cases} r(e, v) \text{ or } r(v', v) \\ \neg r(e, v) \text{ or } \neg r(v', v) \end{cases} \tag{12}$$

with $e \in \mathcal{E}, v, v' \in \{v_1, \ldots, v_N\},$ and $r \in \mathcal{R}$.

T-norms. Neuro-symbolic methods for CQA incorporate logical operators in their computations by relaxing them into continuous functions known as **t-norms** and **t-conorms** from fuzzy logics [33], where values in $[0, 1]$ model degrees of truth. In this setting, conjunction ($\wedge$) is generalized by a *t-norm* $\top : [0, 1]^2 \to [0, 1]$, disjunction ($\vee$) by a *t-conorm* $\perp : [0, 1]^2 \to [0, 1]$, and negation ($\neg$) by the standard fuzzy complement $1 - x$. Using these relaxations, a query $q(v_1, \ldots, v_N)$ in disjunctive normal form can be expressed as

$$q(v_1, \ldots, v_N) = (c_1^1 \top \ldots \top c_{m_1}^1) \perp \ldots \perp (c_1^n \top \ldots \top c_{m_n}^n), \tag{13}$$

where each atom c_j^i is mapped to a fuzzy truth value in $[0, 1]$.

An example is the *product* t-norm and its dual t-conorm, defined as

$$\top(x, y) = x \cdot y, \tag{14}$$

$$\perp(x, y) = x + y - x \cdot y, \tag{15}$$

which recover classical Boolean logic when $x, y \in \{0, 1\}$. Several other t-norms exist, for an in-depth description see Hájek [13].

Fuzzy vectors for CQA. Several neuro-symbolic methods assign a vector $\mathbf{v}_i \in [0,1]^{|\mathcal{E}|}$ to each variable v_i in a query, which represents the likelihood that the variable is assigned each entity in $\mathcal{E}$ [1–3, 35, 7, 32]. Similarly, constants in a query can be represented as one hot vectors $\mathbf{c} \in \{0,1\}^{|\mathcal{E}|}$, such that $\mathbf{c}[e] = 1$ for a constant entity e, and all other entries are zero. This vector can then be transformed with t-norms, t-conorms, and negations depending on the query being answered.

Projections and negations. For a query formula $q(v_1, v_2) = r(v_1, v_2)$ with $r \in \mathcal{R}$, a fuzzy vector $\mathbf{v}_1$ is projected via a function $f_r(\mathbf{v}_1) = \mathbf{v}_2 \in [0,1]^{|\mathcal{E}|}$ which yields the fuzzy vector for v_2. Often the function f_r is a parameterized neural network optimized for 1-hop link prediction [1–3] or CQA [35, 11]. For a negation $q(v_1, v_2) = \neg r(v_1, v_2)$, the fuzzy vector is computed as $\mathbf{1} - f_r(\mathbf{v}_1)$, where $\mathbf{1}$ is an all-ones vector of length $|\mathcal{E}|$.

Conjunctions and disjunctions. For a conjunction $q(v_1, v_2, v_3) = r_1(v_1, v_2) \wedge r_2(v_3, v_2)$ with $r_1, r_2 \in \mathcal{R}$ the two predicates are first projected as $f_{r1}(\mathbf{v}_1)$ and $f_{r2}(\mathbf{v}_3)$, and the fuzzy vector of $\mathbf{v}_2$ is computed via the t-norm: $\mathbf{v}_2 = f_{r1}(\mathbf{v}_1) \top f_{r1}(\mathbf{v}_3)$. For a disjunction $r_1(v_1, v_2) \wedge r_2(v_3, v_2)$ the result is computed via the t-conorm, $\mathbf{v}_2 = f_{r1}(\mathbf{v}_1) \perp f_{r1}(\mathbf{v}_3)$.

A.2 Proofs of properties of SCORE

Proposition 1 (Monotonicity of SCORE updates)**.** *For any two entities $e_1, e_2 \in \mathcal{E}$, suppose their preference contributions are equal:*

$$w_p \Delta(P_s^+, e_1) - w_n \Delta(P_s^-, e_1) = \; w_p \Delta(P_s^+, e_2) - w_n \Delta(P_s^-, e_2). \tag{10}$$

Then the relative ordering of their fuzzy scores is preserved under the SCORE update:

$$\mathbf{v}_j[e_1] < \mathbf{v}_j[e_2] \implies \mathbf{v}_j^{new}[e_1] < \mathbf{v}_j^{new}[e_2].$$

Proof. By definition of the update in Equation (9),

$$\mathbf{v}_j^{\text{new}}[e] = \sigma(\text{logit}(\mathbf{v}_j[e]) + \delta(e)) \,,$$

where $\delta(e) = w_p \Delta(P_s^+, e) - w_n \Delta(P_s^-, e)$. If $\delta(e_1) = \delta(e_2)$, then the same shift is applied to both $\text{logit}(\mathbf{v}_j[e_1])$ and $\text{logit}(\mathbf{v}_j[e_2])$. Since $\sigma(\cdot)$ is strictly monotone, the ordering of the original logits is preserved, and so is the ordering of the values in the fuzzy vector. Hence SCORE is monotonic with respect to the scores of the underlying model: it preserves the base model's ranking whenever preference contributions are equal. □

The monotonicity of SCORE contributes to interpretability: whenever the score of entity is promoted or demoted, the exact source of this shift can be traced back to $\Delta(P_s^+, e)$ and $\Delta(P_s^-, e)$. This makes it transparent whether the change is due to evidence from positively or negatively labeled examples, providing a direct explanation of how similarity constraints affect the final scores.

Proposition 2 (Linear complexity of SCORE)**.** *The computational complexity of applying SCORE is $O(|\mathcal{E}|)$, i.e. linear with respect to the number of entities.*

Proof. Let $|\mathcal{E}|$ be the number of entities and k the number of examples in the preference set. For each candidate entity $e \in \mathcal{E}$, SCORE computes normalized similarities $\overline{\text{sim}}(e, e_i)$ with all k labeled examples, which costs $O(kd)$ if embeddings are d-dimensional. Aggregating these values and computing the update in Equation (9) also requires $O(k)$. Thus the cost per entity is $O(kd)$, and the total cost is $O(|\mathcal{E}|kd)$. Since k is small (on the order of tens of examples) and d is fixed by the underlying embedding model, both factors are constant in practice. Therefore the overall complexity scales linearly with $|\mathcal{E}|$. □

Connection with posterior log-odds. Assuming a probabilistic interpretation of scores, we reinterpret the similarity constraint $s(e)$ (previously a binary predicate) as the probability that an entity $e \in \mathcal{E}$ is to satisfy the constraint. For brevity, let $\mathbf{v} = \mathbf{v}_j[e]$ denote the base CQA score for entity e. The SCORE update can then be written as

$$\mathbf{v}^{\text{new}} = \sigma(\text{logit}(\mathbf{v}) + \text{logit}(s(e))) \,. \tag{16}$$

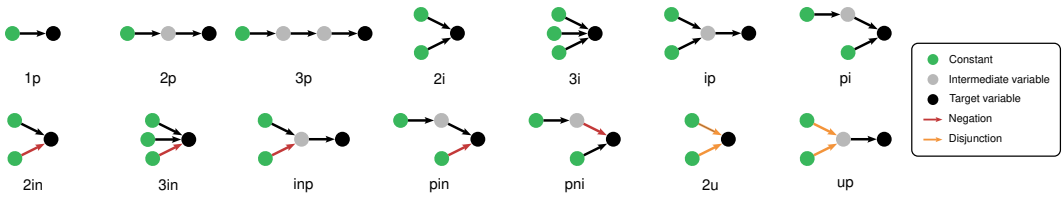


Figure 6: Query graphs of different types covered in the datasets used in our experiments, along with their designated name.

Table 3: Statistics of the datasets used in our experiments.

	Knowledge Graph			Queries			Preference Sets		
Dataset	Entities	Relations	Edges	Train	Validation	Test	Train	Validation	Test
FB15k237	14,505	237	310,079	157,479	20,961	21,352	580,623	68,834	70,365
Hetionet	45,158	24	2,250,198	91,820	29,475	29,488	325,443	99,785	99,284
NELL995	63,361	200	142,804	75,446	17,289	17,435	245,511	52,446	52,589

Taking logits gives

$$\operatorname{logit}(\mathbf{v}^{\text{new}}) = \operatorname{logit}(\mathbf{v}) + \operatorname{logit}(s(e)), \tag{17}$$

and by the definition of the logit,

$$\log \frac{\mathbf{v}^{\text{new}}}{1-\mathbf{v}^{\text{new}}} = \log \frac{\mathbf{v}}{1-\mathbf{v}} + \log \frac{s(e)}{1-s(e)}. \tag{18}$$

This is equivalent to the additive update of log-odds in Bayesian inference: the posterior log-odds equal the prior log-odds plus the log-likelihood ratio of the evidence. More concretely, let x be the random variable indicating whether e is an answer. If we view $\mathbf{v} = P(x = 1)$ as the prior from the CQA model and $s(e)$ as a data-dependent score that encodes the strength of evidence for $x = 1$ versus $x = 0$, then

$$\log \frac{P(x=1 \mid \text{data})}{P(x=0 \mid \text{data})} = \log \frac{P(x=1)}{P(x=0)} + \log \frac{P(\text{data} \mid x=1)}{P(\text{data} \mid x=0)}. \tag{19}$$

Thus, the SCORE update mirrors the Bayesian log-posterior update when scores are interpreted probabilistically. While the base CQA model and similarity function may not produce calibrated probabilities, operating in logit space ensures that preference evidence is combined additively and propagated consistently with the logical operators of the query.

A.3 Datasets

In this section we provide additional details on how we construct a generalized benchmark for SimCQA with similarity constraints on arbitrary variables.

Base resources. We start from standard benchmarks for complex query answering [25], which cover 14 query types involving conjunction, disjunction, and negation (illustrated in Figure 6). For each knowledge graph (Hetionet, FB15k-237, and NELL995), we generate training, validation, and test queries following the same procedure as in prior work. Each query is associated with a designated *target variable* v_i whose answers form the set $A_i(q)$.

Preference sets and constrained answers. Following Daza et al. [10], we rely on clusterings of textual embeddings as a scalable proxy for unobserved similarity constraints. In their human validation subset on FB15k-237, these embedding-derived clusters agree with human judgments in 79% of cases, supporting their use as preference sets for large-scale evaluation. For each query, we (i) identify a variable v_j (which may be the target or any intermediate variable), (ii) collect the entities that appear as valid bindings for v_j and compute an embedding of their textual descriptions, and (iii) apply hierarchical agglomerative clustering to obtain two partitions of this set. These partitions define preference sets of the form $P_s = \{(e, \texttt{true}) : e \in C^+\} \cup \{(e, \texttt{false}) : e \in C^-\}$, where C^+ and

C^- are clusters. The induced constrained answer set $\hat{A}_i(q, s, j)$ is then computed by propagating the partition of v_j through the query, yielding a corresponding partition of the target answers. Textual information is used only to construct preference sets and constrained answer sets, and the evaluated SimCQA methods only access graph topology.

Text-based entity embeddings. Embeddings for entities used during the clustering stage are computed using their textual description. We use pretrained text embedding models available through Hugging Face[1] to derive vector representations of entities from their textual descriptions. For FB15k237, we adopt the `NovaSearch/stella_en_400M_v5` model[2], which produces 1,024-dimensional embeddings. For Hetionet, we rely on `Alibaba-NLP/gte-Qwen2-7B-instruct`[3]. At the time of writing, this model holds the top position on the Massive Text Embedding Benchmark[4] for medical-domain tasks and yields embeddings of dimension 3,584.

Generalization protocol. A key difference with Daza et al. [10] is how we split answers across $\mathcal{T}^{\text{train}}$, $\mathcal{T}^{\text{valid}}$, and $\mathcal{T}^{\text{test}}$. In their original setup, preference examples could overlap with answers reachable in $\mathcal{T}^{\text{test}}$, meaning that at test time the model effectively observes part of the ground truth answers (a few-shot learning scenario). To ensure stricter generalization, we impose the following rules:

- The ground truth constrained answer set $\hat{A}_i(q, s, j)$ is defined only with respect to $\mathcal{T}^{\text{test}}$. This ensures that evaluation reflects the model's ability to generalize to unseen facts.
- Preference sets P_s are restricted to contain only entities reachable via $\mathcal{T}^{\text{train}} \cup \mathcal{T}^{\text{valid}}$. This prevents test leakage and forces the model to extrapolate similarity constraints to novel cases at test time.

Constraints on arbitrary variables. Whereas Daza et al. [10] considered constraints only on the target variable, we generalize the construction to allow constraints on *any* variable in the query. This is achieved by clustering the bindings of each free variable v_j in the query (not just the target), and retaining those partitions that induce non-trivial and non-overlapping partitions of the target answer set. This extension produces preference/evidence pairs where feedback is given on intermediate variables, leading to richer scenarios for similarity-constrained reasoning.

Filtering and quality control. To ensure meaningful preference sets, we apply the following filters: (i) only variables with at least 10 and at most 100 distinct bindings are considered for clustering; (ii) both positive and negative clusters must induce at least 5 valid answers on the training edges; (iii) induced answer sets must be non-empty and disjoint. These criteria discard degenerate cases and guarantee that similarity constraints provide useful supervisory signals.

Output. Each dataset instance consists of:

1. a query q and target variable v_i,
2. the variable v_j on which preferences are applied,
3. a preference set P_s built from clusters of bindings of v_j,
4. the ground truth constrained answer set for the target variable v_i: $\hat{A}_i(q, s, j)$ derived from $\mathcal{T}^{\text{test}}$.

We repeat this process across all query types and variables, producing a diverse benchmark that covers both biomedical and encyclopedic domains. Dataset statistics are reported in Table 3 and Table 4.

A.4 Metrics

Global ranking quality. For each entity in the unconstrained answer set $A_i(q)$, we determine its rank r, and compute the Mean Reciprocal Rank (MRR) and Hits@k (H@k), computed per answer as $\text{MRR} = \frac{1}{r}$ and $\text{H@k} = \mathbb{1}[r \leq k]$ (where $\mathbb{1}$ is an indicator function) and averaged over all queries.

[1] https://huggingface.co/

[2] https://huggingface.co/NovaSearch/stella_en_400M_v5

[3] https://huggingface.co/Alibaba-NLP/gte-Qwen2-7B-instruct

[4] https://huggingface.co/spaces/mteb/leaderboard

Table 4: Statistics of the queries and preference sets in the datasets used in our experiments.

Structure	1p	2p	3p	2i	3i	ip	pi	2in	3in	inp	pin	pni	2u	up	Total
	FB15k237														
	Training														
Queries	7,596	34,720	56,495	16,205	8,870	0	0	5,300	3,821	9,384	8,891	6,197	0	0	157,479
Pref. Sets	19,075	121,420	240,770	46,302	26,265	0	0	16,536	11,530	41,128	38,217	19,380	0	0	580,623
	Validation														
Queries	1,631	1,746	2,255	1,341	753	1,249	1,501	1,210	1,560	2,184	1,825	782	1,087	1,837	20,961
Pref. Sets	4,149	5,967	8,876	3,972	2,275	4,099	5,239	3,516	4,549	8,140	6,256	2,182	3,251	6,363	68,834
	Test														
Queries	1,934	1,776	2,267	1,347	814	1,337	1,399	1,191	1,553	2,133	1,844	814	1,080	1,863	21,352
Pref. Sets	4,898	6,157	9,138	3,875	2,398	4,358	4,834	3,456	4,581	8,017	6,516	2,356	3,206	6,575	70,365
	Hetionet														
	Training														
Queries	19,595	20,075	2,000	20,075	20,075	0	0	2,000	2,000	2,000	2,000	2,000	0	0	91,820
Pref. Sets	66,076	79,488	9,127	68,553	66,012	0	0	7,281	6,768	7,503	7,586	7,049	0	0	325,443
	Validation														
Queries	9,975	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	29,475
Pref. Sets	32,653	5,185	5,805	5,024	5,017	5,040	5,801	5,158	5,068	5,223	4,970	4,829	4,877	5,135	99,785
	Test														
Queries	9,988	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	29,488
Pref. Sets	32,865	5,049	5,796	5,056	4,980	4,843	5,783	5,092	5,121	5,218	4,789	4,630	4,820	5,242	99,284
	NELL995														
	Training														
Queries	1,871	17,649	24,859	4,547	3,034	0	0	3,848	2,459	5,824	6,016	5,339	0	0	75,446
Pref. Sets	4,677	50,328	92,914	11,221	7,530	0	0	11,507	7,151	22,040	22,145	15,998	0	0	245,511
	Validation														
Queries	942	1,281	1,532	824	468	836	1,042	1,381	1,383	2,126	1,859	1,058	1,152	1,405	17,289
Pref. Sets	2,254	4,043	5,159	2,292	1,291	2,410	2,855	3,945	4,214	7,229	6,038	3,046	3,270	4,400	52,446
	Test														
Queries	1,004	1,253	1,607	993	635	772	1,039	1,344	1,510	2,106	1,770	938	1,106	1,358	17,435
Pref. Sets	2,437	3,917	5,514	2,659	1,789	2,077	2,933	3,860	4,479	7,182	5,743	2,618	3,076	4,305	52,589

Similarity constraint satisfaction. Here we measure whether methods rank preferred answers higher than non-preferred ones. We partition the answer set $A_i(q)$ into answers in the constrained answer set $\hat{A}_i(q, s, j)$, which we denote as $A_i(q)^+$, and those not in it, denoted as $A_i(q)^-$. We then compute Pairwise Accuracy:

$$\text{PA} = \sum_{e^+ \in A_i(q)^+} \sum_{e^- \in A_i(q)^-} \mathbb{1}\left[r(e^+) < r(e^-)\right], \tag{20}$$

where $r(e)$ indicates the ranking of entity e.

We also compute the Normalized Discounted Cumulative Gain at k (NDCG@k), assigning 0 relevance to non-answers, 1 to answers in $A_i(q)^-$, and 2 to answers in $A_i(q)^+$. Concretely, the discounted cumulative gain is

$$\text{DCG@}k = \sum_{e \in A_i(q)} \frac{2^{\text{rel}(e)} - 1}{\log_2(j+1)}, \tag{21}$$

where

$$\text{rel}(e) = \begin{cases} 0 & \text{if } e \notin A_i(q) \\ 1 & \text{if } \in A_i(q) \setminus \hat{A}_i(q, s, j) \\ 2 & \text{if } \in \hat{A}_i(q, s, j) \end{cases} \tag{22}$$

and the normalized score is

$$\text{NDCG@}k = \frac{\text{DCG@}k}{\text{IDCG@}k}, \tag{23}$$

where $\text{IDCG@}k$ is the maximum achievable DCG under the given relevance scores. Since NDCG@k is normalized by the ideal ranking, a value of 1 indicates that all preferred answers appear above non-preferred answers, which in turn appear above non-answers. We report the average of PA and NDCG@k over all queries.

A.5 Experimental details

Confidence intervals. In all plots with shaded regions, we report 95% confidence intervals for each preference-set size $t = 1, \ldots, 10$. Concretely, we first compute the mean and standard error of the mean (SEM) across queries, and then multiply the latter by the two-sided Student-t critical value with $n - 1$ degrees of freedom, where n is the number of evaluated queries. This corresponds to $\mathrm{CI}_{95} = \mathrm{SEM} \cdot t_{0.975,n-1}$ after scaling metrics to percentages.

Compute resources. All experiments were run on a cluster using a single NVIDIA A100 GPU with 40GB of memory, 120GB RAM, and an Intel Xeon CPU.

Licenses of existing assets. FB15k-237 is distributed under the CC-BY-4.0 license, and Hetionet is distributed under CC0. NELL995 is freely available, but to the best of our knowledge there is no explicit license information for it.

SCORE. The only two parameters that need to be tuned in SCORE are the weights w_p and w_n in Equation (9) that control the strength of shifts in logit space to the base score due to similarities with the preference set. We run a grid search with values in $\{0.25, 0.5, 1.0\}$, for a total of 9 possible combinations of w_p and w_n.

NQR. We perform a grid search with values of the learning rate in $\{1 \times 10^{-5}, 1 \times 10^{-4}\}$, the margin in the preference loss in $\{0.05, 0.1, 0.25\}$, and the KL divergence weight in $\{0.1, 1.0, 10\}$, resulting in a total of 18 possible combinations of hyperparameters. As done by Daza et al. [10], we train NQR on queries of type 1p, as training on more query types is more expensive but brings little benefits.

MeanCosine. The update rule in the MeanCosine baseline is

$$\mathbf{v}_i^{\text{new}}[e] = \alpha\, \mathbf{v}_i[e] + (1 - \alpha)\left(\frac{1+\beta}{2|P_s^+|}\sum_{e_i \in P_s^+} \mathrm{sim}(e_i, e) - \frac{1-\beta}{2|P_s^-|}\sum_{e_i \in P_s^-} \mathrm{sim}(e_i, e)\right). \tag{24}$$

Intuitively, MeanCosine computes a convex combination of the original score, and a score due to the mean of raw cosine similarity values, which is balanced with the hyperparameter $\alpha \in [0, 1]$. Similarities from P_s^+ are added, and those from P_s^- are subtracted, and the balance of these two is determined by the value of $\beta \in [-1, 1]$. We run a grid search for the values of α in $\{0.25, 0.5, 0.75\}$, and $\beta \in \{-0.5, 0, 0.5\}$ for a total of 9 possible configurations. In general SimCQA, the result of this update at an intermediate variable is no longer guaranteed to lie in $[0, 1]$, so we pass it through a sigmoid before it is consumed by subsequent fuzzy logical operators. We select its hyperparameters with the same grid search.

A.6 Additional results

We present more detailed results on SimCQA performance for each of the 14 types of complex queries we consider in our work (shown in Figure 6).

Following our main experiments, we apply a similarity constraint to a query by using preference sets of size $t = 1, \ldots, 10$. For each value of t we then compute NDCG@10 and report the average over the 10 sizes. We present results for target-variable SimCQA in Table 11 and for general SimCQA in Table 12.

For general SimCQA, we suffix the query type with sℓ, where ℓ is the zero-based index of the variable on which the similarity constraint is applied. Variables are ordered in a topological order from the leaf nodes to the root of the query graph (see Figure 6). For example, for a 2p chain, the intermediate variable corresponds to index 0 (2ps0), and the terminal variable corresponds to index 1 (2ps1).

Additional metrics. We present additional results on more fine-grained ranking metrics, including H@3, in Figure 7 and Figure 8.

Numeric results. To facilitate direct comparison, Tables 5 to 10 report the values shown in Figures 3 and 4 for preference sets of size $t \in \{1, 5, 10\}$.

Table 5: Numeric results for target-variable SimCQA on FB15k-237 at preference-set sizes $t \in \{1, 5, 10\}$.

Method	PA			MRR			H@1			H@10			NDCG@10		
	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$
Base CQA model	54.7	54.7	54.7	**24.2**	**24.2**	**24.2**	**16.3**	**16.3**	**16.3**	**40.0**	40.0	40.0	33.4	33.4	33.4
MeanCosine	63.0	66.6	67.2	22.9	23.7	23.9	15.5	16.0	16.1	37.7	39.1	39.5	32.7	34.0	34.3
NQR	**70.6**	**74.0**	**75.1**	18.9	19.9	20.1	12.5	13.2	13.3	31.8	33.3	33.8	28.9	30.8	31.3
SCORE	58.6	60.2	60.3	24.0	24.1	**24.2**	16.1	16.2	16.2	**40.0**	**40.1**	**40.1**	**34.1**	**34.4**	**34.4**

Table 6: Numeric results for target-variable SimCQA on Hetionet at preference-set sizes $t \in \{1, 5, 10\}$.

Method	PA			MRR			H@1			H@10			NDCG@10		
	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$
Base CQA model	51.7	51.7	51.7	**32.5**	**32.5**	**32.5**	**25.1**	25.1	25.1	**47.2**	**47.2**	**47.2**	**41.5**	41.5	41.5
MeanCosine	57.3	59.3	59.8	30.9	32.0	32.4	24.3	**25.2**	**25.4**	43.7	45.5	46.2	40.2	42.0	42.5
NQR	**80.1**	**85.4**	**86.6**	20.9	21.8	22.0	14.9	15.8	16.0	33.0	34.0	34.2	33.2	35.4	35.9
SCORE	56.9	58.0	58.1	30.7	31.8	32.4	23.0	24.4	25.1	45.9	46.6	46.9	41.1	**42.4**	**42.9**

Alternative base CQA models. We present further results on both target-variable and general SimCQA when replacing the base CQA models with GNN-QE [35] and UltraQuery [11] in Figure 9 and Figure 10. We observe the same qualitative behavior previously reported with QTO: incorporating similarity constraints with SCORE consistently improve preference satisfaction while preserving or improving global ranking performance, even under changes of the base CQA models.

A.7 Robustness to noisy preferences

To simulate imperfect feedback, we independently flip the label of a fraction $k \in \{0, 0.1, 0.2, 0.4, 0.6, 0.8\}$ of the entities in each preference set, so that preferred entities are treated as non-preferred and vice versa. As shown in Figure 11, noise mainly affects pairwise accuracy: at $t = 10$, it decreases from 58.4 without noise to 55.0 at $k = 0.4$, still above the base CQA model (53.3), and falls below it only when feedback becomes predominantly misleading (50.9 at $k = 0.8$). In contrast, MRR changes only slightly, from 20.9 to 20.8 at $k = 0.8$. SCORE therefore tolerates moderate noise while largely preserving global answer quality. Regarding imbalanced preference sets, SCORE averages logit similarities within P_s^+ and P_s^- separately, so that a larger set does not dominate the update merely by containing more examples.

A.8 Alignment between embeddings and preferences

SCORE approximates similarity constraints with entity embeddings from the base CQA model, so its effectiveness depends on how well the embedding space reflects a preference. To measure this, for each preference set we build an average-linkage hierarchical clustering of its entities $P_s^+ \cup P_s^-$ using the cosine similarity of the ComplEx embeddings of QTO, and identify the smallest cluster S containing all preferred entities P_s^+. We define its precision as $|P_s^+|/|S|$, which equals 1 if the embedding space separates preferred from non-preferred entities perfectly. We compute it for the full preference sets of target-variable SimCQA on the FB15k-237 test set, and average it per query.

Precision ranges from 0.16 to 1.0, and is perfect for only 1.3% of the queries. For 29% of the queries, precision is below 0.5, meaning that S contains more non-preferred than preferred entities. This occurs most often for 3i (35%) and 2i (34%) queries, which are also the query types with the smallest gains in pairwise accuracy (+3.06 and +3.34, respectively, see Table 13). Precision is positively associated with the pairwise accuracy of SCORE, with a Spearman correlation of 0.20 overall and between 0.15 and 0.29 for every query type (all significant at $p < 0.05$ after Benjamini–Hochberg correction). By contrast, its correlation with MRR is close to zero overall (-0.03), and significant only for 1p, 2i, and 3i queries. When the embedding space is poorly aligned with a preference, SCORE is thus less able to capture it, while global ranking quality is not systematically affected.

A.9 Multiple simultaneous constraints

We evaluate SCORE with similarity constraints applied simultaneously to the first two variables of 2p and 3p queries on FB15k-237 (648 and 504 test queries, respectively). Following the protocol

Table 7: Numeric results for target-variable SimCQA on NELL-995 at preference-set sizes $t \in \{1, 5, 10\}$.

Method	PA			MRR			H@1			H@10			NDCG@10		
	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$
Base CQA model	51.7	51.7	51.7	**16.5**	16.5	16.5	**8.7**	8.7	8.7	**32.2**	32.2	32.2	26.7	26.7	26.7
MeanCosine	56.4	58.8	59.1	14.7	15.8	16.3	8.2	8.7	9.0	27.9	30.3	31.1	24.6	26.5	27.2
NQR	**63.7**	**63.8**	**63.8**	9.6	10.1	10.2	4.8	5.1	5.1	18.9	19.9	20.2	18.1	19.0	19.2
SCORE	53.9	54.3	54.3	16.3	**16.8**	**17.0**	**8.7**	**9.1**	**9.3**	31.9	**32.5**	**32.7**	**26.8**	**27.7**	**27.8**

Table 8: Numeric results for general SimCQA on FB15k-237 at preference-set sizes $t \in \{1, 5, 10\}$.

Method	PA			MRR			H@1			H@10			NDCG@10		
	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$
Base CQA model	53.3	53.3	53.3	**20.9**	**20.9**	**20.9**	**13.9**	**13.9**	**13.9**	**35.0**	**35.0**	**35.0**	31.7	31.7	31.7
SCORE	56.5	58.2	58.4	20.8	**20.9**	**20.9**	13.7	13.8	13.8	**35.0**	34.9	**35.0**	**32.3**	**32.6**	**32.7**
MeanCosine	**58.6**	**61.1**	**61.4**	17.3	17.8	17.9	11.4	11.7	11.8	28.7	29.7	30.0	26.1	27.0	27.2

in Section A.3, we first cluster the bindings of the first variable, and then cluster the bindings of the second variable that are reachable from the preferred cluster of the first. Preferred answers are those reached only through bindings that satisfy both constraints. In each interaction step, one example is added to the preference set of each variable, and SCORE applies each update as the corresponding fuzzy vector is computed. Table 14 reports the results. As in single-constraint settings, SCORE substantially improves pairwise accuracy and NDCG@10, while MRR remains close to the base CQA model.

Table 9: Numeric results for general SimCQA on Hetionet at preference-set sizes $t \in \{1, 5, 10\}$.

Method	PA			MRR			H@1			H@10			NDCG@10		
	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$
Base CQA model	51.1	51.1	51.1	**26.1**	**26.1**	**26.1**	**19.7**	**19.7**	**19.7**	**38.5**	**38.5**	**38.5**	**36.0**	36.0	36.0
SCORE	**55.3**	56.6	56.8	24.7	25.7	**26.1**	18.2	19.3	**19.7**	37.6	38.2	**38.5**	**36.0**	**37.3**	**37.7**
MeanCosine	55.1	**56.7**	**57.0**	24.4	25.2	25.5	18.9	19.5	**19.7**	35.0	36.4	36.9	33.4	34.8	35.2

Table 10: Numeric results for general SimCQA on NELL-995 at preference-set sizes $t \in \{1, 5, 10\}$.

Method	PA			MRR			H@1			H@10			NDCG@10		
	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$	$t=1$	$t=5$	$t=10$
Base CQA model	51.4	51.4	51.4	**15.4**	15.4	15.4	**8.2**	8.2	8.2	**30.0**	30.0	30.0	26.5	26.5	26.5
SCORE	53.5	54.2	54.3	15.3	**15.7**	**15.9**	**8.2**	**8.5**	**8.6**	29.7	**30.3**	**30.5**	**26.7**	**27.4**	**27.6**
MeanCosine	**55.1**	**57.0**	**57.2**	12.8	13.6	14.0	7.0	7.4	7.7	24.1	26.0	26.6	22.7	24.1	24.6

Table 11: NDCG@10 results for **target-variable SimCQA**, averaged over preference sets of size $t = 1, \ldots, 10$.

Method	1p	2in	2i	2p	2u	3in	3i	3p	inp	ip	pin	pi	pni	up	Avg.
FB15k237															
Base CQA model	43.90	23.34	47.40	28.75	34.37	37.14	50.54	27.16	21.48	34.32	20.05	41.32	9.87	27.98	31.97
MeanCosine	44.25	23.76	47.92	29.67	34.80	37.20	50.92	26.95	**22.81**	34.29	20.38	40.53	10.33	28.34	32.30
NQR	41.61	21.13	46.13	25.24	32.33	33.82	49.43	21.30	19.48	27.79	16.92	36.12	9.25	25.95	29.04
SCORE	**44.43**	**24.07**	**48.06**	**29.84**	**34.91**	**37.80**	**51.38**	**28.34**	22.62	**35.61**	**21.20**	**42.03**	**10.57**	**28.73**	**32.83**
Hetionet															
Base CQA model	52.95	33.11	44.51	17.10	44.21	22.11	48.94	14.58	13.63	18.38	11.27	34.94	12.84	18.18	27.62
MeanCosine	**53.18**	33.51	44.62	19.42	**44.62**	23.56	47.97	13.56	**17.76**	20.54	**14.35**	31.66	13.35	18.76	28.35
NQR	45.24	24.65	39.93	14.65	35.57	19.20	44.64	11.75	13.20	13.67	10.96	26.11	11.23	16.13	23.35
SCORE	52.79	**33.59**	**45.93**	**20.69**	44.31	**24.23**	**50.38**	**15.96**	17.27	**23.53**	14.23	**37.55**	**14.41**	**19.52**	**29.60**
NELL995															
Base CQA model	47.02	23.36	**41.79**	29.45	33.53	**22.78**	47.10	23.35	21.30	29.71	15.70	31.72	10.34	29.79	29.07
MeanCosine	46.98	22.99	41.74	28.85	32.81	21.74	47.36	22.64	22.23	27.67	14.60	28.97	10.17	29.35	28.43
NQR	37.55	14.81	33.77	20.28	26.46	13.23	37.64	14.15	15.04	21.62	7.94	23.04	6.25	21.19	20.93
SCORE	**47.58**	**23.99**	41.74	**30.43**	**33.63**	22.59	**47.57**	**24.81**	**22.98**	**31.32**	**16.54**	**33.38**	**11.02**	**30.51**	**29.86**

Table 12: NDCG@10 results for **general SimCQA**, averaged over preference sets of size $t = 1, \ldots, 10$.

Method	1ps0	2ins0	2is0	2ps0	2ps1	2us0	3ins0	3is0	3ps0	3ps1	3ps2	inps0	inps1	ips0	ips1	pins0	pins1	pis0	pis1	pnis1	ups0	ups1	Avg.
FB15k237																							
Base CQA model	43.90	23.34	47.40	29.29	28.75	34.37	37.14	50.54	28.60	27.72	27.16	22.15	21.48	40.29	34.32	21.48	20.05	36.97	41.32	9.87	30.34	27.98	31.11
MeanCosine	44.25	23.76	47.92	12.74	29.67	34.80	37.20	50.92	15.72	12.09	26.95	8.55	**22.81**	14.34	34.29	9.23	20.38	20.19	40.53	10.33	13.41	28.34	25.38
SCORE	**44.43**	**24.07**	**48.06**	**29.39**	**29.84**	**34.91**	**37.80**	**51.38**	**29.00**	**28.36**	**28.34**	**22.65**	22.62	**40.42**	**35.61**	**22.36**	**21.20**	**38.12**	**42.03**	**10.57**	**30.68**	**28.73**	**31.84**
Hetionet																							
Base CQA model	52.95	33.11	44.51	17.14	17.10	44.21	22.11	48.94	32.02	14.59	14.58	13.52	13.63	21.42	18.38	13.84	11.27	30.27	34.94	12.84	17.18	18.18	24.85
MeanCosine	**53.18**	33.51	44.62	8.43	19.42	**44.62**	23.56	47.97	28.39	9.01	13.56	7.96	**17.76**	11.87	20.54	3.04	**14.35**	**31.58**	31.66	13.35	7.62	18.76	22.94
SCORE	52.79	**33.59**	**45.93**	**21.22**	**20.69**	44.31	**24.23**	**50.38**	**32.30**	**20.55**	**15.96**	**16.62**	17.27	**26.58**	**23.53**	**17.16**	14.23	30.31	**37.55**	**14.41**	**18.59**	**19.52**	**27.17**
NELL995																							
Base CQA model	47.02	23.36	**41.79**	30.98	29.45	33.53	**22.78**	47.10	25.64	23.55	23.35	23.26	21.30	37.64	29.71	25.20	15.70	38.29	31.72	10.34	27.57	29.79	29.05
MeanCosine	46.98	22.99	41.74	16.10	28.85	32.81	21.74	47.36	21.69	13.37	22.64	11.89	22.23	22.86	27.67	14.21	14.60	**40.49**	28.97	10.17	16.04	29.35	25.21
SCORE	**47.58**	**23.99**	41.74	**31.45**	**30.43**	**33.63**	22.59	**47.57**	**26.78**	**25.07**	**24.81**	**24.52**	**22.98**	**39.30**	**31.32**	**26.21**	**16.54**	40.30	**33.38**	**11.02**	**27.63**	**30.51**	**29.97**

Table 13: Pairwise accuracy (PA) and MRR per query type for **target-variable SimCQA** on FB15k-237, averaged over preference sets of size $t = 1, \ldots, 10$. Δ denotes the difference between SCORE and the base CQA model.

Method	1p	2in	2i	2p	2u	3in	3i	3p	inp	ip	pin	pi	pni	up	Avg.
Pairwise accuracy															
Base CQA model	58.15	56.03	54.72	55.59	57.70	52.98	54.22	53.01	54.65	55.55	53.62	55.50	57.39	53.97	55.22
MeanCosine	63.36	67.45	58.95	69.59	66.01	61.64	57.60	70.98	69.95	68.18	71.60	64.55	69.81	67.07	66.20
NQR	79.45	75.37	69.22	73.89	74.47	70.45	67.28	72.66	72.37	75.71	73.72	72.71	75.79	71.78	73.20
SCORE	62.82	62.44	58.06	61.04	62.12	58.92	57.28	58.23	59.58	60.75	60.42	59.15	62.34	58.75	60.14
Δ	+4.67	+6.41	+3.34	+5.46	+4.42	+5.94	+3.06	+5.21	+4.92	+5.20	+6.80	+3.65	+4.94	+4.78	+4.91
MRR															
Base CQA model	35.01	16.19	34.13	21.09	27.11	22.99	34.80	19.49	14.51	25.95	13.96	29.79	6.94	20.07	23.00
MeanCosine	34.75	15.87	34.06	20.67	26.91	22.69	34.78	17.63	14.32	24.15	12.99	28.27	6.81	19.51	22.39
NQR	29.34	13.51	30.52	16.38	23.78	19.35	31.45	12.91	11.51	17.90	10.18	23.45	5.86	16.69	18.77
SCORE	34.53	16.06	34.13	21.13	26.86	22.93	35.11	19.55	14.76	25.97	14.16	29.93	7.06	20.04	23.02
Δ	-0.48	-0.13	+0.00	+0.04	-0.25	-0.06	+0.31	+0.06	+0.25	+0.02	+0.20	+0.14	+0.12	-0.03	+0.01

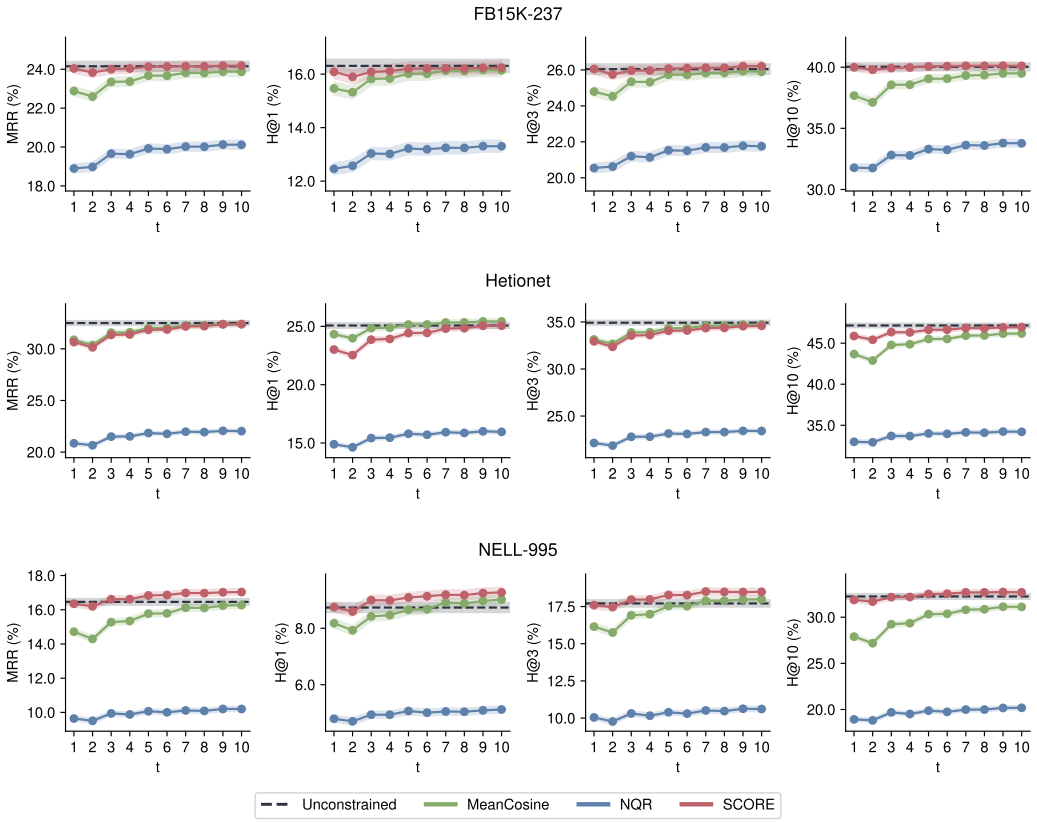


Figure 7: Additional metrics for SimCQA with similarity constraints on the **target** variable. Shaded areas indicate a 95% confidence interval.

Table 14: Results with similarity constraints on two variables of 2p and 3p queries (FB15k-237), averaged over preference sets of size $t = 1, \ldots, 10$.

	PA		MRR		NDCG@10	
Method	2p	3p	2p	3p	2p	3p
Base CQA model	52.54	49.16	**18.07**	11.54	26.76	25.64
SCORE	**58.17**	**54.48**	17.96	**11.58**	**27.88**	**26.23**

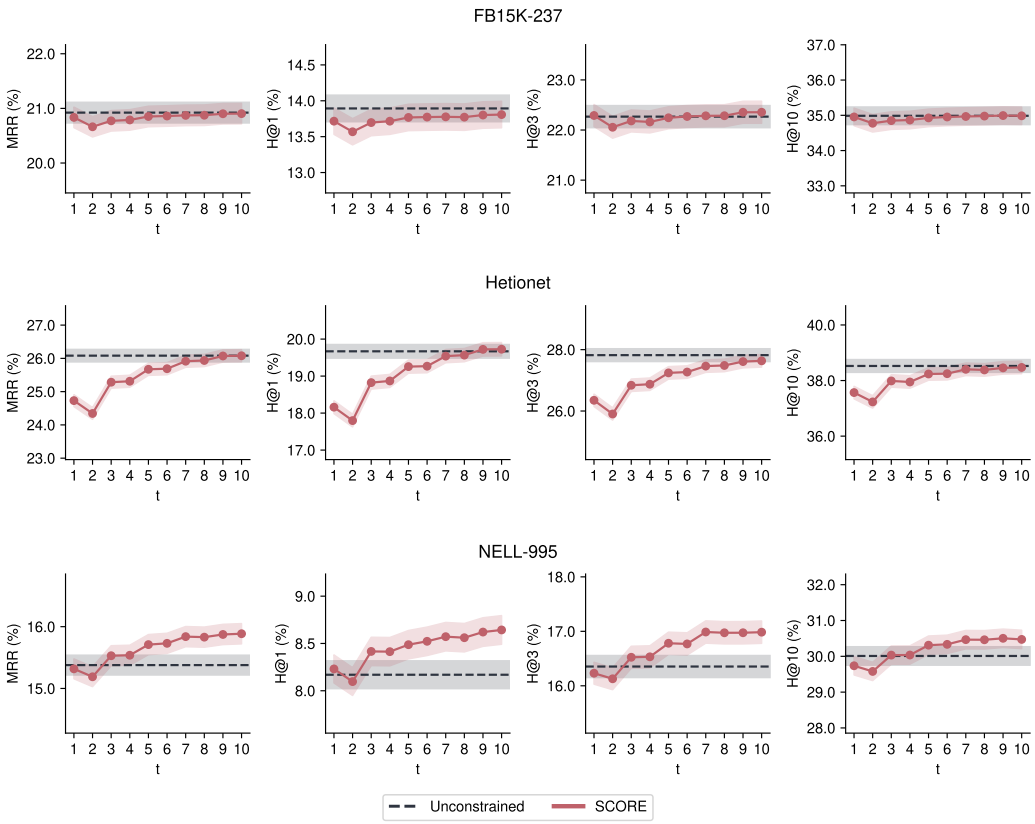


Figure 8: Additional metrics for SimCQA with similarity constraints on the **target and intermediate** variables. Shaded areas indicate a 95% confidence interval.

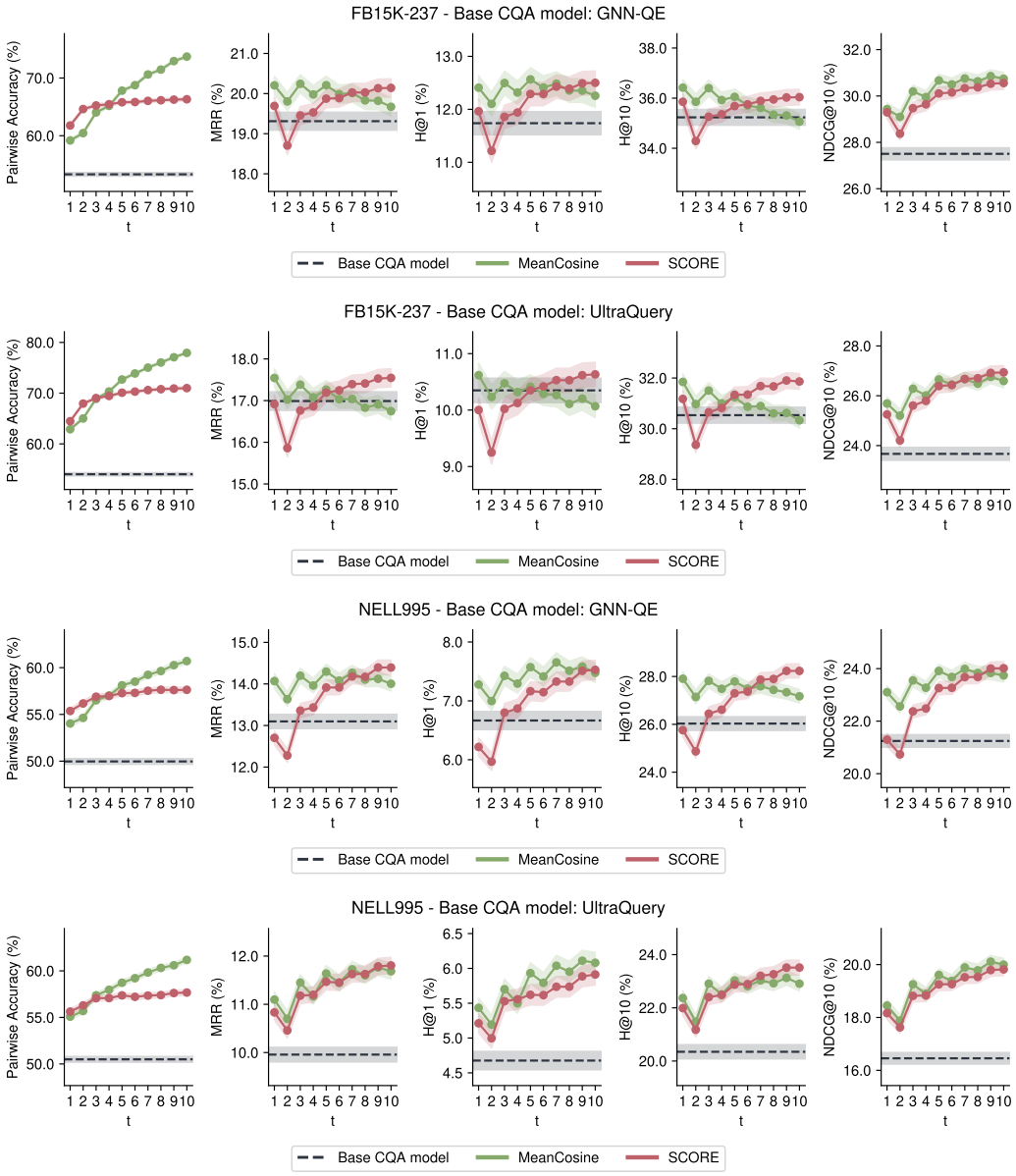


Figure 9: SimCQA results for similarity constraints on the **target** variable, using different base CQA models: GNN-QE [35] and UltraQuery [11]. Shaded areas indicate a 95% confidence interval.

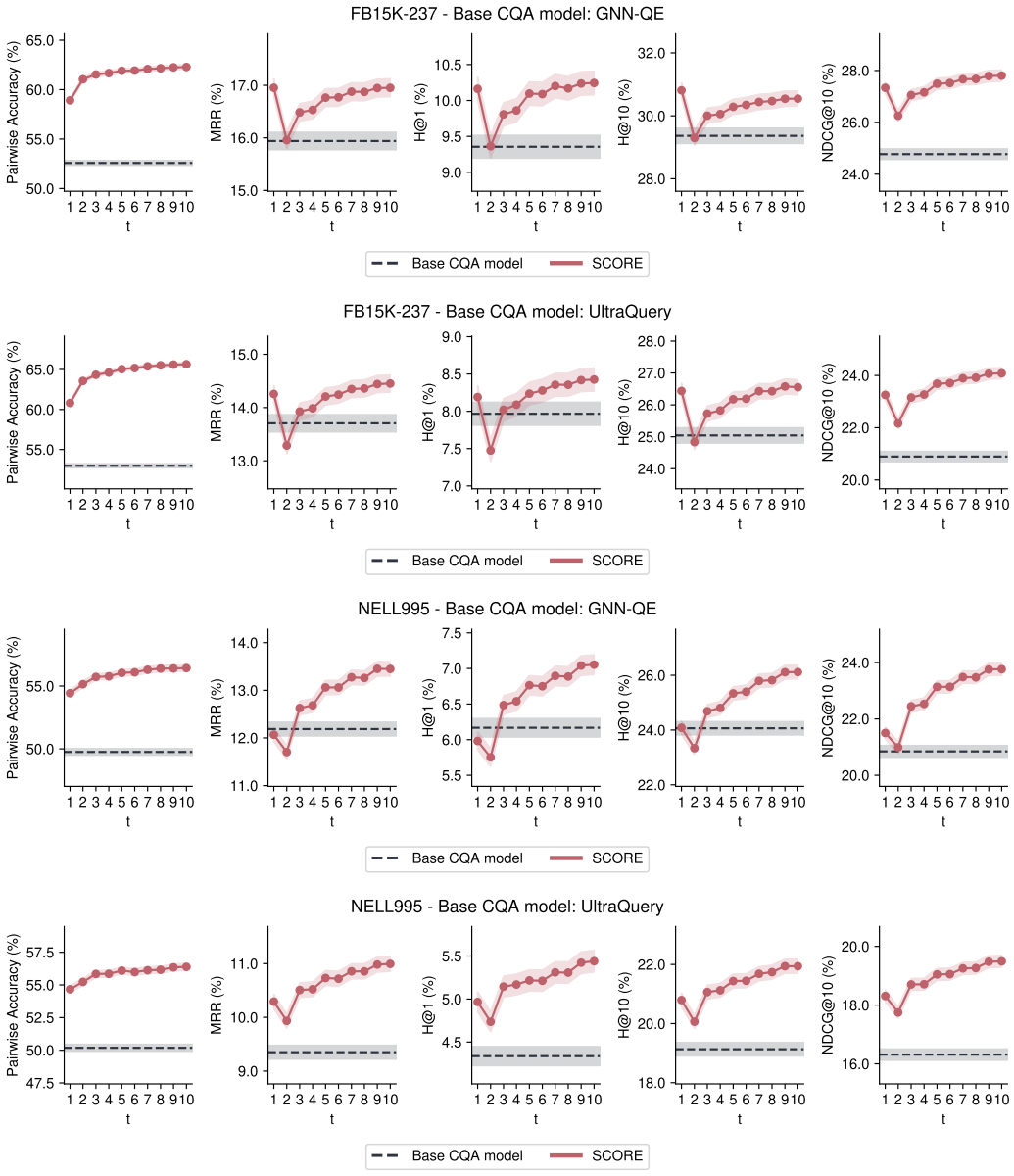


Figure 10: SimCQA results for similarity constraints on **target and intermediate** variables, using different base CQA models: GNN-QE [35] and UltraQuery [11]. Shaded areas indicate a 95% confidence interval.

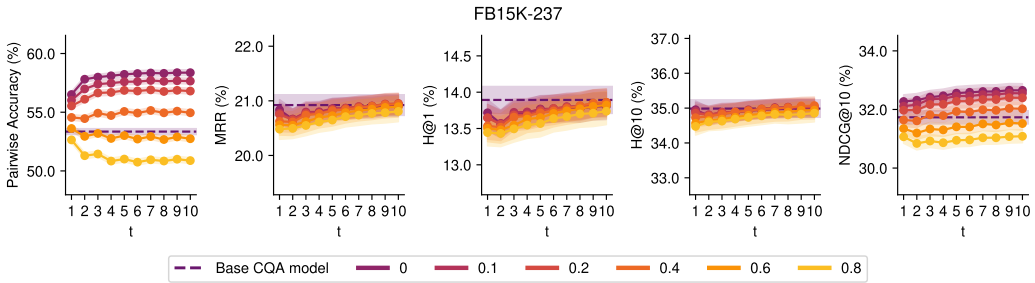


Figure 11: SCORE for general SimCQA on FB15k-237 when a fraction k of the labels in each preference set is flipped. Shaded areas indicate a 95% confidence interval.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions and scope of the paper, including the SimCQA formulation, the SCORE method, and the experimental evaluation. These claims are supported by the methodological sections, theoretical properties, and experiments in the main paper and appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The paper includes a dedicated paragraph on limitations and future work (Section 6).

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper states the assumptions for the theoretical propositions and provides complete proofs in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: The paper describes the datasets, evaluation protocol, baselines, metrics, query settings, and hyperparameter selection, with additional details in the appendix. An anonymized implementation link is also provided in the abstract.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provides a link to a public repository, which includes the code, data access details, and instructions needed to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The main paper and appendix specify the train/validation/test split setting, datasets, baselines, metrics, and hyperparameter searches. The appendix further details dataset construction and experimental settings needed to interpret the reported results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main result figures report 95% confidence intervals, and the appendix specifies how they are computed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: The appendix specifies the hardware used for the experiments, including the GPU, CPU, and available memory. The main paper also reports per-query runtime overhead for SCORE.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: To the best of our knowledge, the research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper does not include a dedicated discussion of broader societal impacts. As a methodological paper on knowledge-graph query answering, such impacts are possible but are not analyzed in the current manuscript.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk generative models, scraped data, or similarly sensitive assets that would require special misuse safeguards. It focuses on a query-answering method evaluated on established knowledge-graph benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper credits the datasets and prior methods through citations, and the appendix states the dataset licenses where available.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces new assets in the form of an implementation and expanded benchmark instances, and their documentation is described in the manuscript and the accompanying code repository.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or experiments with human subjects. All evaluations are conducted on existing knowledge-graph datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects or crowdsourcing. As a result, IRB approval or an equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of the proposed method or experiments.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.