Certainty in Uncertainty: Reasoning over Uncertain Knowledge Graphs with Statistical Guarantees

Yuqicheng Zhu[1,2]*, Jingcheng Wu[1]*, Yizhen Wang[1], Hongkuan Zhou[1,2], Jiaoyan Chen[3], Evgeny Kharlamov[2,4], Steffen Staab[1,5]
[1]University of Stuttgart, [2]Bosch Center for AI,
[3]The University of Manchester, [4]University of Oslo, [5]University of Southampton
yuqicheng.zhu@de.bosch.com

Abstract

Uncertain knowledge graph embedding (UnKGE) methods learn vector representations that capture both structural and uncertainty information to predict scores of unseen triples. However, existing methods produce only point estimates, without quantifying predictive uncertainty—limiting their reliability in high-stakes applications where understanding confidence in predictions is crucial. To address this limitation, we propose UNKGCP, a framework that generates prediction intervals guaranteed to contain the true score with a user-specified level of confidence. The length of the intervals reflects the model's predictive uncertainty. UNKGCP builds on the conformal prediction framework but introduces a novel nonconformity measure tailored to UnKGE methods and an efficient procedure for interval construction. We provide theoretical guarantees for the intervals and empirically verify these guarantees. Extensive experiments on standard benchmarks across diverse UnKGE methods further demonstrate that the intervals are sharp and effectively capture predictive uncertainty. To support future research on this topic, we release our code[1].

1 Introduction

Knowledge graphs (KGs) represent factual knowledge as triples of the form $\langle$ *Head Entity*, *Predicate*, *Tail Entity*$\rangle$, capturing relationships between real-world entities (Hogan et al., 2021). Knowledge in KGs can be uncertain due to noise and errors from inaccurate automated extraction processes (Pujara et al., 2013), or because some facts are inherently probabilistic, such as molecular interactions (Szklarczyk et al., 2016). To capture such uncertainty, uncertain KGs (UnKGs) associate each triple with a score that reflects the likelihood of the fact being true (Wu et al., 2012; Speer et al., 2017a; Mitchell et al., 2018a).

Reasoning over UnKGs aims to predict the score of unseen triples, leveraging the structure and uncertainty information encoded in the observed graph. Existing approaches (Chen et al., 2019, 2021b,a; Zhou et al., 2024) extend KG embedding (KGE) techniques to UnKGs, which we refer to as UnKGE methods. Specifically, these methods represent entities and predicates as numerical vectors, assess the plausibility of triples based on distance (Bordes et al., 2013) or dot product (Nickel et al., 2011), and then map this plausibility to a score in the range $[0, 1]$.

However, existing UnKGE models produce only point estimates without capturing how confident the model is in its predictions. In real-world applications (Zhou et al., 2025; Sadikaj et al., 2025), especially in high-stakes domains, it is crucial to know the range within which the true score is likely to fall. For example, when predicting the likelihood of a harmful drug interaction based on a biomedical KG, a point estimate of 0.3 might suggest low risk. However, if the model also indicated that plausible scores range from 0.2 to 0.95, it would reveal high uncertainty, indicating that further investigation is needed before taking clinical action.

To the best of our knowledge, no existing method provides a statistically grounded way to quantify uncertainty in the predictions of UnKGE methods. We take the first step toward addressing this gap by introducing UNKGCP, a framework that applies *conformal prediction* (Vovk et al., 2005) to quantify uncertainty through a prediction interval—a set of plausible values that is guaranteed to contain the ground truth with a user-specified confidence level. The core idea is to assess how "atypical" a candidate prediction is compared to previously seen data using a *nonconformity score*. Based on these scores, the method selects a threshold that ensures the constructed interval includes the ground truth with the desired level of confidence. Specifically, we introduce **a novel nonconformity measure tailored to**

*Equal contribution.
[1]https://github.com/0sidewalkenforcer0/UnKGCP

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 8742–8764
November 4-9, 2025

UnKGE methods that allows the prediction intervals to adapt to the difficulty of each query, along with **an efficient procedure for constructing such intervals**.

We provide theoretical guarantees on the coverage of the ground truth by the prediction intervals (Proposition 1) and validate our approach through extensive experiments on commonly used UnKG benchmarks across a range of UnKGE methods. Our empirical study shows that: (1) UNKGCP produces prediction intervals that both satisfy the theoretical guarantees and remain sharp and informative; (2) the intervals adapt to query-specific uncertainty; (3) UNKGCP is sample-efficient, achieving similar performance using only about 20% of the calibration set.

2 Related Work

UnKGE Methods. Several UnKGE methods have been proposed to support reasoning under triple-level uncertainty (Chen et al., 2019, 2021b,a). As the first in this line of research, UKGE (Chen et al., 2019) extends DistMult (Yang et al., 2015a) to UnKGs by mapping plausibility scores to the $[0, 1]$ range, replacing the loss function with mean squared error, and augmenting training data using probabilistic soft logic. PASSLEAF (Chen et al., 2021b) generalizes this framework to support a broader range of KGE backbones and improves negative sampling by predicting scores for negative triples using semi-supervised learning. In contrast to vector-based models, Chen et al. (2021a) represent entities as boxes and encode relations as affine transformations between boxes, achieving improved performance and robustness to noise. Another line of work addresses reasoning under schema-level uncertainty. For example, Zhu et al. (2023, 2024b) approximate probabilistic inference in statistical $\mathcal{EL}$ using box embeddings.

Uncertainty Quantification in KGE. As highlighted by Zhu et al. (2024a), the predictions of KGE models can vary substantially with minor changes to training conditions (e.g., random seed), underscoring the importance of uncertainty quantification in KGE. Some recent work has explored this: Tabacof and Costabello (2020); Safavi et al. (2020) apply post-hoc calibration techniques to map plausibility scores from KGEs into calibrated probabilities. However, uncertainty quantification in the context of UnKGE has been largely overlooked. Zhu et al. (2024b) produce intervals via ensemble methods, but these intervals lack formal statistical guarantees.

Conformal Prediction. This work applies conformal prediction, a general framework for uncertainty quantification with finite-sample statistical guarantees. Conformal prediction has been applied across various domains, including image classification (Angelopoulos et al., 2021a), natural language processing (Maltoudoglou et al., 2020; Campos et al., 2024), node classification and regression on graphs (Huang et al., 2024; Zargarbashi et al., 2023; Zargarbashi and Bojchevski, 2023), and link prediction on deterministic KGs (Zhu et al., 2025b,a).

Among these, Zhu et al. (2025b) and Zhu et al. (2025a) are most closely related to our work, but they focus on link prediction in deterministic KGs and adopts nonconformity measures and set construction procedure specifically designed for that context. These methods cannot be directly applied to UnKGE tasks due to fundamental differences in output space and objectives: link prediction involves ranking a finite set of candidate entities and yields discrete prediction sets to quantify uncertainty, whereas score prediction in UnKGE requires constructing real-valued prediction intervals for continuous outputs. As a result, the design of nonconformity scores, set construction, and theoretical guarantees differs substantially.

3 Preliminaries

3.1 Uncertain Knowledge Graph Embeddings

Let E and R represent finite sets of *entities* and *predicates*, respectively. A KG is a subset of $E \times R \times E$, where each element, called *triple*, represents a fact. An UnKG extends KG by associating each fact with a confidence score indicating the likelihood of the fact being true. Formally, an UnKG can be defined as a set of *weighted triples*:

$$\big\{\langle h, r, t, c\rangle \mid \langle h, r, t\rangle \in E \times R \times E, c \in [0, 1]\big\}.$$

An UnKGE model is a function $M_\theta : E \times R \times E \to [0, 1]$ that assigns confidence scores to triples. The parameters θ of the model are learned by minimizing the discrepancy between predicted and ground truth confidence scores. A typical training objective is the mean squared error (Chen et al., 2019):

$$L = \sum_{(q,c)\in\mathcal{T}\cup\mathcal{T}^-} |M_\theta(q) - c|^2, \qquad (1)$$

where $q = \langle h, r, t \rangle$ is a query triple, and, $\mathcal{T}, \mathcal{T}^-$ denote the sets of positive and negative training examples, respectively.

Note that UnKGs often do not contain explicit negative examples. Negative triples are commonly generated by corrupting positive triples, for example, by replacing the head or tail entity with a randomly selected entity from E (Chen et al., 2019). However, confidence scores for these negative triples are not simply assigned a value of 0. Instead, Chen et al. (2019) employ probabilistic soft logic to estimate their scores, while Chen et al. (2021b) adopt a semi-supervised learning framework for this purpose.

3.2 Conformal Prediction

In this section, we recall essential concepts from conformal prediction as introduced in Vovk et al. (2005). Consider a dataset $Z = \{(x_i, y_i)\}_{i=1}^n$, with inputs $x_i \in \mathcal{X}$ and the corresponding labels $y_i \in \mathcal{Y}$. We denote the space of individual examples by $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, and the space of all possible example sets by $\mathcal{Z}^*$.

3.2.1 Confidence Predictor

Given a test input x_{n+1}, our goal is to design an algorithm Γ that, instead of predicting a single label for y_{n+1}, outputs a *prediction interval*—a subset of $\mathcal{Y}$ that contains the true label with a specified *confidence level* $\alpha \in [0, 1]$. To reflect the trade-off between confidence and informativeness, the prediction intervals are required to expand as α increases: intuitively, achieving higher confidence necessitates including more possible labels.

Formally, a *confidence predictor* is a measurable function

$$\Gamma : \mathcal{Z}^* \times \mathcal{X} \times [0, 1] \to 2^{\mathcal{Y}}, \tag{2}$$

that maps a set of (training) examples, a test input, and a desired confidence level to a subset of possible labels. In our case, this subset corresponds to an interval. For notational convenience, we write $\Gamma^\alpha(Z, x) := \Gamma(Z, x, \alpha)$ to denote the prediction interval at level α. We require the confidence predictor to satisfy the following *monotonicity property*:

$$\Gamma^{\alpha_1}(Z, x_{n+1}) \subseteq \Gamma^{\alpha_2}(Z, x_{n+1}), \forall \alpha_1 \leq \alpha_2. \tag{3}$$

The quality of a confidence predictor is evaluated based on three key **desiderata**: *validity*, *efficiency*, and *conditionality*.

- **Validity** ensures that, in the long run, the prediction interval covers the true label with probability at least α. Formally, the coverage of a confidence predictor Γ at level α is defined as

$$Cov(\Gamma^\alpha) := \mathop{\mathbb{P}}_{\substack{Z \sim \mathcal{P}^n, \\ (x_{n+1}, y_{n+1}) \sim \mathcal{P}}} \Big(y_{n+1} \in \Gamma^\alpha(Z, x_{n+1}) \Big), \tag{4}$$

  where $\mathcal{P}$ denotes the (unknown) joint distribution over examples. We say that Γ is *exactly valid* if $Cov(\Gamma^\alpha) = \alpha$, and *conservatively valid* if $Cov(\Gamma^\alpha) \geq \alpha$.

- **Efficiency** refers to the tightness of the prediction intervals. Given the same confidence level, a more efficient confidence predictor produces sharper (i.e., more informative) prediction intervals.

- **Conditionality** expresses the degree to which the confidence predictor adapts to the difficulty of individual examples. Ideally, the size of the prediction interval should reflect how uncertain the model is about the specific input x_{n+1}: smaller for easy cases and larger for hard ones.

3.2.2 Conformal Predictor

A *conformal predictor* is a confidence predictor that provides rigorous validity guarantees. It leverages a *nonconformity measure* $S : \mathcal{Z}^* \times \mathcal{Z} \to \mathbb{R}$, which quantifies how "strange" a test example appears relative to observed examples. Given a model $\hat{f}_Z : \mathcal{X} \to \mathcal{Y}$ trained on Z, a common nonconformity measure for regression tasks is the absolute residual:

$$S\big(Z, (x, y)\big) = \Big| \hat{f}_Z(x) - y \Big|. \tag{5}$$

Applying S to an example z yields a *nonconformity score* $s = S(Z, z)$, which reflects how atypical z appears when compared against the examples in Z.

To generate a prediction interval for a test input x_{n+1}, the conformal predictor proceeds as follows. For each candidate $y \in \mathcal{Y}$, it forms an augmented dataset $Z' = Z \cup (x_{n+1}, y)$ and computes nonconformity scores for all examples in Z'. In particular, it computes

$$\begin{aligned} s_i &:= S(Z', z_i), i = 1, \ldots, n, \\ s_{n+1} &:= S\big(Z', (x_{n+1}, y)\big). \end{aligned} \tag{6}$$

The label y is included in the prediction interval if its nonconformity score s_{n+1} is not among the largest $1-\alpha$ fraction of scores in Z', that is:

$$\Gamma^{\alpha}_{\mathrm{CP}}(Z, x_{n+1}) := \Big\{ y \in \mathcal{Y} : \frac{|\{i = 1, \ldots, n+1 : s_i \geq s_{n+1}\}|}{n+1} > 1 - \alpha \Big\}. \quad (7)$$

By constructing prediction intervals as described above, all conformal predictors have the following validity guarantees.

Theorem 1 (Vovk et al. (2005), Lei et al. (2018))**.** *Assume the examples in Z and the test example z_{n+1} are independent and identically distributed (i.i.d). For any confidence level $\alpha \in [0, 1]$ and any nonconformity measure S, the conformal predictor $\Gamma^{\alpha}_{\mathrm{CP}}$ is conservatively valid:*

$$\mathbb{P}\big(y_{n+1} \in \Gamma^{\alpha}_{\mathrm{CP}}(Z, x_{n+1})\big) \geq \alpha. \quad (8)$$

Furthermore, if $\{s_i\}_{i=1}^{n}$ contains no ties, $\Gamma^{\alpha}_{\mathrm{CP}}$ is also asymptotically exactly valid:

$$\lim_{n \to \infty} \mathbb{P}\big(y_{n+1} \in \Gamma^{\alpha}_{\mathrm{CP}}(Z, x_{n+1})\big) = \alpha. \quad (9)$$

Remark 1. *The validity guarantees of conformal prediction hold under the even weaker assumption of* exchangeability *(Shafer and Vovk, 2008; Vovk et al., 2005). Exchangeability allows dependencies among examples, as long as their joint distribution remains invariant under permutations.*

4 Conformalized Uncertain Knowledge Graph Embeddings (UnKGCP)

Conformal prediction is a general uncertainty quantification framework requiring careful adaptation for specific tasks via tailored nonconformity measures and efficient prediction interval constructions. In this section, we introduce **an efficient way to construct prediction intervals**, analyse its time complexity and prove its validity guarantees. Moreover, we propose **a novel nonconformity measure** designed for UnKGE models, ensuring query-specific prediction intervals.

4.1 Problem Setup

We consider a set of weighted triples $\mathcal{T} = \{tr_i\}_{i=1}^{n}$, where each $tr_i = (q_i, c_i)$ consists of a query triple $\langle h, r, t \rangle$ and an associated confidence score $c_i \in [0, 1]$. Given an UnKGE model $M_{\mathcal{T}}$ trained on $\mathcal{T}$, the reasoning task on UnKGs is to predict the confidence score for a test query q_{n+1}. We aim to quantify the uncertainty in model predictions by constructing prediction intervals $\Gamma^{\alpha}(\mathcal{T}, q_{n+1})$ at a use-specified confidence level $\alpha \in [0, 1]$, satisfying the properties as described in Section 3.2.1.

4.2 Efficient Set Construction

Applying conformal prediction as described in Section 3.2.2 to UnKGE requires examining infinitely many potential confidence scores $c \in [0, 1]$ and training a new UnKGE model for each query-potential value pair (q_{n+1}, c), which is computationally prohibitive.

To overcome this, we employ inductive conformal prediction (ICP) (Section 4.2 Vovk et al., 2005; Lei et al., 2018) to construct prediction intervals efficiently and avoid examining infinitely many cases. Specifically, we randomly partition $\mathcal{T}$ into two disjoint sets: a *proper training set* $\mathcal{T}_{\text{train}} = \{tr_i\}_{i=1}^{m}$ and a *calibration set* $\mathcal{T}_{\text{cal}} = \{tr_i\}_{i=m+1}^{n}$ of size $\ell = n - m$. An UnKGE model is trained exclusively on $\mathcal{T}_{\text{train}}$, after which it remains fixed to compute nonconformity scores on $\mathcal{T}_{\text{cal}}$ and new queries.

Given a nonconformity measure S and a user-specified confidence level $\alpha \in [0, 1]$, the ICP-based prediction interval for a test query q_{n+1} is defined as

$$\Gamma^{\alpha}_{\mathrm{ICP}}(\mathcal{T}, q_{n+1}) := \Big\{ c \in [0, 1] : \frac{|\{i = m+1, \ldots, n+1 : s_i \geq s_{n+1}\}|}{\ell + 1} > 1 - \alpha \Big\}, \quad (10)$$

where

$$s_i := S\big(\mathcal{T}_{\text{train}}, tr_i\big), i = m+1, \ldots, n, \quad (11)$$
$$s_{n+1} := S\big(\mathcal{T}_{\text{train}}, (q_{n+1}, c)\big),$$

4.2.1 Time Complexity

This procedure avoids repeated model retraining and significantly improves computational efficiency. Given k test queries, the overall computational complexity scales as

$$\mathcal{O}\Big(T_{\text{train}} + (\ell + k)T_{\text{infer}} + \ell \log \ell + k \log \ell\Big), \quad (12)$$

where T_{train} is the one-time cost of training the UnKGE model. For mainstream UnKGE methods $T_{\text{train}} = \mathcal{O}(|E|d)$, with d denoting the embedding dimension. $T_{\text{infer}} = \mathcal{O}(d)$ is the time to compute the confidence score for a query. We allocate time $\ell \log \ell$ to sort the nonconformity scores obtained from the calibration set $\mathcal{T}_{\text{cal}}$, $\log \ell$ to determine

the rank of s_{n+1} among the scores in $\{s_i\}_{i=m+1}^{n}$. Since only $kT_{\text{infer}} = \mathcal{O}(kd)$ and $k \log \ell$ depend on k, while T_{train}, ℓT_{infer}, and $\ell \log \ell$ are constant in k, the overall complexity as $k \to \infty$ is $\mathcal{O}\big(k(d + \log \ell)\big)$—**asymptotically linear** in k.

When scaling to larger graphs, the dominant cost is T_{train}. Since T_{infer} scales only linearly with the embedding dimension d. All other terms are completely independent of the graph size. Thus, once the UnKGE model is trained, **our method runs with a computational cost that is agnostic to the graph size**.

4.2.2 Validity Guarantees

We show that the ICP-based conformal predictor retains the formal validity guarantees stated in Theorem 1, while benefiting from a more efficient set construction process. The proof is provided in Appendix A.

Proposition 1. *Assume weighted triples in $\mathcal{T}$ and the test weighted triple (q_{n+1}, c_{n+1}) are i.i.d. For any confidence level $\alpha \in [0, 1]$ and any nonconformity measure S, the ICP-based conformal predictor $\Gamma^{\alpha}_{\text{ICP}}$ is conservatively valid:*

$$\mathbb{P}\big(c_{n+1} \in \Gamma^{\alpha}_{\text{ICP}}(\mathcal{T}, q_{n+1})\big) \geq \alpha. \tag{13}$$

Furthermore, if $\{s_i\}_{i=m+1}^{n}$ contains no ties, $\Gamma^{\alpha}_{\text{ICP}}$ is also asymptotically exactly valid:

$$\lim_{\ell \to \infty} \mathbb{P}\big(c_{n+1} \in \Gamma^{\alpha}_{\text{ICP}}(\mathcal{T}, q_{n+1})\big) = \alpha. \tag{14}$$

4.3 Adaptive Nonconformity Measures

While the previous section provides validity guarantees for any nonconformity measure, standard choices such as the absolute residual (Equation (5)) lead to prediction intervals of fixed width, regardless of the uncertainty in individual test queries—thus failing to satisfy the conditionality desideratum.

To formalize this, let us denote the set of sorted nonconformity scores from the calibration set as

$$\{s_{m+1}, s_{m+2}, \ldots, s_n\} = \{s_{(1)}, s_{(2)}, \ldots, s_{(\ell)}\}, \tag{15}$$

with $s_{(1)} \leq s_{(2)} \leq \cdots \leq s_{(\ell)}$. Then Equation (10) can be equivalently expressed as

$$\Gamma^{\alpha}_{\text{ICP}}(\mathcal{T}, q_{n+1}) := \big\{c \in [0, 1] : s_{n+1} \leq s_{(\lceil \alpha(\ell+1) \rceil)}\big\}. \tag{16}$$

By definition, $s_{n+1} = |M(q_{n+1}) - c|$, which results in a symmetric prediction interval with absolute residual as nonconformity measure:

$$\Gamma^{\alpha}_{\text{ICP}}(\mathcal{T}, q_{n+1}) := \big[M(q_{n+1}) - s_{(\lceil \alpha(\ell+1) \rceil)}, M(q_{n+1}) + s_{(\lceil \alpha(\ell+1) \rceil)}\big], \tag{17}$$

where M is the shorthand notation of a fixed UnKGE model trained on the proper training set. Since $s_{(\lceil \alpha(\ell+1) \rceil)}$ is shared across all test queries, the interval width remains constant and does not reflect query-specific uncertainty.

To address this limitation, we introduce an *entropy-normalized absolute residual* as our nonconformity measure:

$$S\big(\mathcal{T}_{\text{train}}, (q, c)\big) := \Big|\frac{M(q) - c}{H\big(M(q)\big)}\Big|, \tag{18}$$

where the normalization term $H\big(M(q)\big)$ is the entropy of the model's prediction:

$$H\big(M(q)\big) := -M(q) \log M(q) - \big(1 - M(q)\big) \log \big(1 - M(q)\big). \tag{19}$$

For a new query q_{n+1}, our method UnKGCP constructs the following prediction interval:

$$\Gamma^{\alpha}_{\text{UnKGCP}}(\mathcal{T}, q_{n+1}) := \big[M(q_{n+1}) - \epsilon, M(q_{n+1}) + \epsilon\big], \tag{20}$$

where $\epsilon = s_{(\lceil \alpha(\ell+1) \rceil)} \cdot H\big(M(q)\big)$ is the query-specific tolerance.

This nonconformity measure scales the residual by the model's predictive uncertainty, allowing predictions with higher entropy (i.e., lower confidence) to tolerate larger residuals. As a result, the prediction intervals adapt to the local difficulty of each query. Importantly, since conformal prediction ensures validity under the i.i.d. assumption regardless of the specific nonconformity measure, our approach retains its theoretical validity guarantees while producing more informative, adaptive intervals, as supported by the empirical results in Table 1 and Figure 2.

5 Experiments

5.1 Experimental Settings

Datasets. We evaluate our method on three commonly used benchmarks: CN15k, NL27k, and PPI5k. CN15k is a subgraph of ConceptNet (Speer et al., 2017b), a commonsense KG. NL27k is derived from NELL (Mitchell et al., 2018b), an automatically constructed KG from web data. The

		CN15k		PPI5k		NL27k	
		coverage	sharpness ↓	coverage	sharpness ↓	coverage	sharpness ↓
UKGE	FPI	0.80 (0.000)	0.84 (0.002)	0.89 (0.000)	0.67 (0.002)	1.00 (0.000)	0.66 (0.002)
	QR	0.09 (0.005)	0.20 (0.008)	0.41 (0.438)	0.40 (0.375)	0.96 (0.107)	0.99 (0.001)
	CP	0.90 (0.002)	0.88 (0.002)	0.90 (0.001)	**0.16 (0.002)**	0.90 (0.001)	**0.27 (0.006)**
	UnKGCP	0.90 (0.001)	**0.82 (0.003)**	0.90 (0.001)	**0.16 (0.002)**	0.91 (0.003)	0.43 (0.018)
PASSLEAF	FPI	0.39 (0.135)	0.71 (0.002)	0.89 (0.000)	0.68 (0.001)	1.00 (0.000)	0.76 (0.001)
	QR	-	-	-	-	-	-
	CP	0.90 (0.002)	0.86 (0.003)	0.90 (0.001)	0.21 (0.002)	0.90 (0.001)	0.54 (0.008)
	UnKGCP	0.90 (0.002)	**0.84 (0.003)**	0.90 (0.001)	**0.20 (0.002)**	0.90 (0.001)	**0.44 (0.005)**
BEUrRE	FPI	0.79 (0.001)	0.70 (0.018)	0.89 (0.000)	0.69 (0.006)	1.00 (0.000)	0.67 (0.002)
	QR	0.59 (0.010)	0.70 (0.007)	0.90 (0.002)	0.49 (0.001)	0.48 (0.002)	0.86 (0.001)
	CP	0.90 (0.001)	0.86 (0.003)	0.90 (0.003)	**0.25 (0.008)**	0.90 (0.002)	0.42 (0.006)
	UnKGCP	0.90 (0.001)	**0.81 (0.003)**	0.90 (0.002)	0.26 (0.008)	0.90 (0.002)	**0.38 (0.003)**

Table 1: Coverage and sharpness results on test triples across three datasets (CN15k, PPI5k, NL27k). We report the average over 10 trials, with standard deviation shown in parentheses. Coverage values ≥ 0.90 are highlighted in green. Among those, the method achieving the best (i.e., lowest) sharpness is **bolded**. Since QR is not directly applicable within the semi-supervised learning framework, no results are reported for QR in PASSLEAF.

confidence scores in these datasets are interpreted as subjective beliefs, representing the system's internal estimate of how likely a statement is to be true based on prior knowledge or heuristics. PPI5k is a subset of the STRING Protein-Protein Interaction Knowledge Base (Szklarczyk et al., 2017), where scores correspond to statistical probabilities derived from experimental evidence. Dataset statistics are summarized in Table 4, with additional details provided in Appendix B.1.

	CN15k		PPI5k		NL27k	
	MSE ↓	MAE ↓	MSE ↓	MAE ↓	MSE ↓	MAE ↓
UKGE	0.24	0.41	0.01	0.04	0.05	0.11
PASSLEAF	0.24	0.41	0.01	0.03	0.06	0.11
BEUrRE	0.12	0.28	0.01	0.06	0.03	0.12

Table 2: Mean squared error (MSE) and mean absolute error (MAE) of the UnKGE models. We report the mean over 10 trials; the standard deviation is negligible.

UnKGE Backbones. We base our experiments on three representative UnKGE methods: UKGE (Chen et al., 2019), PASSLEAF (Chen et al., 2021b), and BEUrRE (Chen et al., 2021a). Detailed descriptions of each method are provided in Appendix B.2.

Confidence Predictors. We compare our proposed method against three established baseline techniques for constructing prediction intervals: (1) Fisher Prediction Intervals (FPI) (Fisher, 1935), (2) Quantile Regression (QR) (Koenker and Bassett Jr, 1978), and (3) Conformal Prediction (CP) (Vovk et al., 2005) with absolute residuals as the nonconformity measure. Additional details are provided in Appendix B.3.

Evaluation Metrics. We evaluate prediction intervals using two standard metrics: *Coverage* and *Sharpness*. Given a test set $\mathcal{T}_{\text{test}} = \{(q_i, c_i)\}_{i=n+1}^{N}$ of size $k = N - n$, these metrics are formally defined as follows:

Coverage measures the fraction of test queries for which the ground-truth confidence score c_i is covered by the prediction interval:

$$\text{Coverage} = \frac{1}{k} \sum_{i=n+1}^{N} \mathbb{1}\left[c_i \in \Gamma^{\alpha}(\mathcal{T}, q_i) \right], \quad (21)$$

and serves as an empirical estimate of the theoretical coverage probability in Equation (4).

Sharpness quantifies the average length of the prediction intervals. Let $\Gamma^{\alpha}(\mathcal{T}, q_i) = [l_i, u_i]$, where l_i and u_i are the lower and upper bounds. Then,

$$\text{Sharpness} = \frac{1}{k} \sum_{i=n+1}^{N} (u_i - l_i). \quad (22)$$

An effective confidence predictor should achieve coverage at least equal to the target confidence level while maintaining the smallest possible sharpness.

5.2 Analysis of Empirical Validity and Efficiency

We evaluate the empirical performance of confidence predictors in terms of *validity* and *efficiency*,

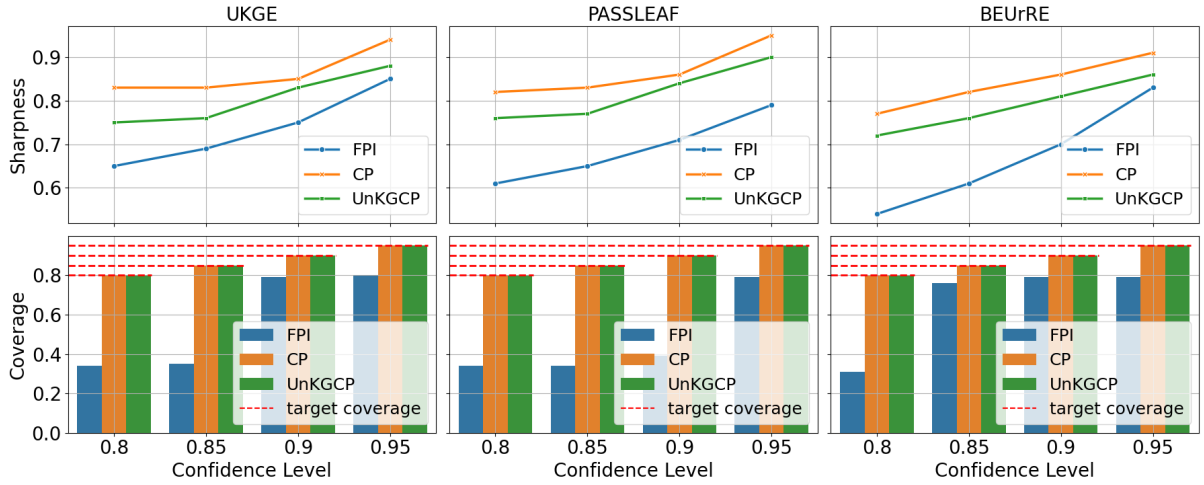

Figure 1: Effect of the confidence level α on the sharpness (top) and coverage (bottom) for test triples on CN15k. Each curve represents one predictor. Red dashed lines indicate the desired coverage levels. Additional results can be found in Figures 4–8 in Appendix C.

using coverage and sharpness as defined in Section 5.1. Table 1 summarizes the results on test triples across the three benchmark datasets and the UnKGE backbones at a 90% confidence level.

The validity guarantees in Proposition 1 are empirically supported, as both conformal predictors (CP and UNKGCP) achieve coverage probabilities closely matching the target confidence across all dataset-backbone configurations. In contrast, other baseline confidence predictors (FISHER and QR) often fail to achieve the target coverage, especially on CN15k and PPI5k. This is likely because FISHER assumes normally distributed residuals—a condition rarely met in UnKGs—and QR relies on a well-specified conditional quantile model, which is challenging given the limited expressiveness of current UnKGE methods.

In terms of efficiency, CP and UNKGCP consistently produce sharper prediction intervals on PPI5k and NL27k. While FISHER and QR yield narrower intervals on CN15k, this comes at the cost of systematic undercoverage. **Notably, our proposed UNKGCP outperforms CP in 7 out of 9 configurations by generating sharper intervals**. Moreover, the average interval lengths from valid confidence predictors (CP and UNKGCP) correlate with UnKGE model performance across datasets (Table 2): wider intervals are associated with higher uncertainty and lower model performance, demonstrating that conformal predictors effectively capture model-level uncertainty through interval length.

In summary, UNKGCP achieves the best overall performance by simultaneously satisfying the validity criterion and generating reasonably sharp prediction intervals. Figure 1 further illustrates the performance of all confidence predictors across multiple confidence levels ranging from 80% to 95% in increments of 5%. The conclusion remains consistent: **UNKGCP maintains superior performance across all confidence levels**. Notably, the length of the prediction intervals increases with higher confidence levels, aligning with the monotonicity property described in Equation (18).

5.3 Analysis of Conditionality

As discussed in Section 4.3, CP produces fixed-length prediction intervals and thus fails to satisfy the conditionality desideratum. In this section, we show that UNKGCP not only produces valid and sharper prediction intervals but also **outperforms CP and other baselines by adapting interval lengths to query difficulty**.

Following Zhu et al. (2025b); Angelopoulos et al. (2021b), we use the *absolute prediction error* as a proxy for instance-level difficulty, with larger errors indicating harder queries. Figure 2 shows that, across all models, average interval length increases with error—demonstrating that prediction intervals adapt to instance difficulty and thus satisfy the conditionality criterion. **This implies that uncertainty can be reliably inferred from the interval length produced by UNKGCP**.

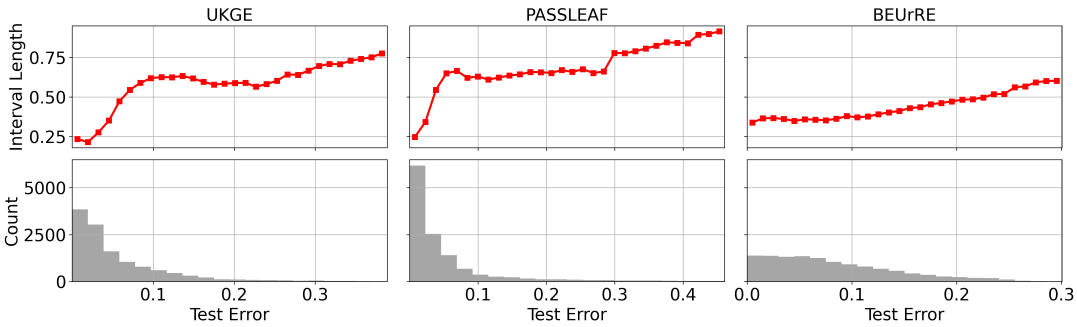

Figure 2: Conditionality analysis on NL27k. Each column corresponds to a different backbone model (BEURRE, UKGE, PASSLEAF). Top: test instances are grouped into 30 bins based on prediction error, and the mean prediction interval length is computed per bin. Only intervals that cover the ground truth are included, as non-covering intervals are not expected to reflect query difficulty. Bottom: histogram of test errors is shown to illustrate their distribution. The complete results are provided in Figures 9–13 in Appendix C.

5.4 Impact of Calibration Set Size

While conformal prediction offers validity guarantees under i.i.d (Vovk et al., 2005; Lei et al., 2018), small calibration sets may yield high-variance estimates of the nonconformity threshold. This can lead to unstable prediction intervals that either under-cover or become unnecessarily wide in practice. In this section, we study how the size of the calibration set influences the performance of UNKGCP in terms of coverage and sharpness.

We randomly sample increasingly larger subsets of the calibration set—starting from 10 triples and doubling the size each time (i.e., 10, 20, 40, ...)—until the full set is used. For each subset size, we repeat the sampling 10 times and report the mean and standard deviation of coverage and sharpness. Figure 3 summarizes the results for three UnKGE-based models on NL27k.

When the calibration set is small (e.g., less than 5–10% of the data), we observe significant variability in both coverage and sharpness. This is caused by unreliable quantile estimates, as limited calibration data can produce a biased distribution of nonconformity scores. As a result, the prediction intervals either under-cover or become overly conservative. As the calibration size increases, both metrics stabilize rapidly. **Notably, UNKGCP demonstrates strong sample efficiency: using only about 20% of the calibration data is sufficient to achieve reliable and stable performance across all models.**

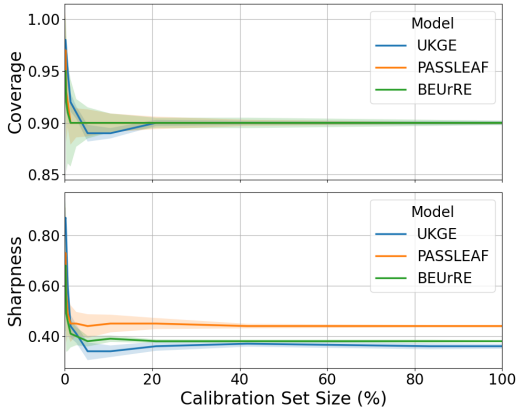

Figure 3: Effect of calibration set size on coverage and sharpness on NL27k. The top panel reports coverage and the bottom panel reports sharpness. In both plots, the lines represent mean values across 10 runs, and the shaded areas indicate standard deviation. The complete results are provided in Figures 14–18 in Appendix C.

6 Sensitivity to Distribution Shift

A key assumption for the validity guarantee in Proposition 1 is that data points are i.i.d. This assumption is easily violated in the negative triple setting, where negatives are synthetically generated by corrupting positives (Chen et al., 2019), leading to distribution shifts across training, calibration, and test sets. We therefore evaluate all methods under this setting.

Table 3 shows that performance on negatives is markedly less stable than on positives. FPI attains 100% coverage with reasonably sharp intervals, likely because nonconformity scores for negatives concentrate near 0, making its Gaussian assumption well-suited here. Nonetheless, **UNKGCP is**

		CN15k		PPI5k		NL27k	
		coverage	sharpness	coverage	sharpness	coverage	sharpness
UKGE	FPI	1.00 (0.000)	**0.22 (0.002)**	1.00 (0.000)	**0.11 (0.001)**	1.00 (0.000)	**0.15 (0.002)**
	QR	0.00 (0.000)	0.20 (0.008)	0.17 (0.312)	0.40 (0.376)	0.70 (0.481)	1.00 (0.001)
	CP	0.82 (0.003)	0.26 (0.002)	0.78 (0.003)	**0.05 (0.002)**	0.79 (0.007)	0.12 (0.004)
	UnKGCP	0.82 (0.003)	**0.26 (0.002)**	0.78 (0.003)	**0.05 (0.001)**	0.79 (0.007)	**0.08 (0.003)**
PASSLEAF	FPI	1.00 (0.000)	**0.11 (0.004)**	1.00 (0.000)	**0.10 (0.005)**	1.00 (0.000)	**0.12 (0.007)**
	QR	-	-	-	-	-	-
	CP	0.75 (0.003)	0.10 (0.003)	0.71 (0.006)	**0.04 (0.001)**	0.70 (0.009)	**0.06 (0.002)**
	UnKGCP	0.75 (0.003)	**0.10 (0.003)**	0.71 (0.008)	0.06 (0.001)	0.70 (0.009)	0.07 (0.002)
BEUrRE	FPI	1.00 (0.000)	**0.29 (0.008)**	1.00 (0.000)	**0.09 (0.004)**	1.00 (0.000)	0.12 (0.005)
	QR	0.58 (0.326)	0.58 (0.160)	0.46 (0.334)	0.09 (0.001)	0.43 (0.074)	0.06 (0.001)
	CP	0.72 (0.006)	0.41 (0.014)	0.75 (0.018)	**0.01 (0.006)**	0.91 (0.001)	0.05 (0.006)
	UnKGCP	0.72 (0.006)	**0.40 (0.014)**	0.75 (0.020)	0.04 (0.006)	0.91 (0.002)	**0.02 (0.002)**

Table 3: Coverage and sharpness results on **negative** test triples across three datasets (CN15k, PPI5k, NL27k). We report the average over 10 trials, with standard deviation shown in parentheses. Coverage values ≥ 0.90 are highlighted in green. Among those, the method achieving the best (i.e., lowest) sharpness is **bolded**. Since QR is not directly applicable within the semi-supervised learning framework, no results are reported for QR in PASSLEAF.

more practical and informative in real-world use. Despite the loss of formal validity under distribution shift, it maintains strong empirical coverage—around 0.8 with UKGE and above 0.7 in most other settings, reaching 0.91 in BOX–NL27k. Crucially, unlike the fixed intervals of FPI, UNKGCP produces query-specific intervals that adapt to prediction difficulty, as illustrated in Figures 9–13.

Another interesting observation, consistent with findings by Kaur et al. (2022), is that our method can be effectively used to detect significant distribution shifts—specifically, in cases where there is a substantial gap between the empirical coverage and the target confidence level.

7 Discussion and Conclusion

We presented UNKGCP, a model-agnostic uncertainty quantification framework for UnKGE models that constructs prediction intervals with formal statistical guarantees (Proposition 1). Experiments across multiple UnKGE models and benchmarks show that UNKGCP produces valid, sharp, and query-adaptive intervals, where interval length reliably reflects predictive uncertainty. Additionally, UNKGCP is sample-efficient, achieving stable performance with only a small calibration set.

Importantly, our uncertainty estimates also offer insights that standard metrics (e.g., mean squared error) fail to capture. For instance, although all UnKGE models seem to achieve reasonably low errors in Table 2 on CN15k, UNKGCP reveals average interval length exceeding 0.8—indicating that the predictions are rather random. This suggests limitations in either the dataset quality or model expressiveness, highlighting the critical role of uncertainty quantification in evaluating model reliability beyond point-based accuracy metrics.

8 Limitations

A limitation of our current method is the assumption that the input graph contains triples annotated with single-valued confidence scores. In practice, however, confidence may be expressed as intervals, or predicted intervals may be added back into the graph. In such cases, the model must be extended to handle interval-valued inputs. Specifically, each input interval can be represented by two components: its mean and its length. The UnKGE model would then be trained to predict both components. During the calibration step of conformal prediction, rather than analyzing the distribution of scalar scores, we would analyze the joint distribution of predicted means and lengths. Separate quantile thresholds would be computed for each, and two conformal intervals—one for the mean and one for the length—would be constructed and subsequently combined to form the final interval, preserving statistical coverage guarantees.

9 Acknowledgments

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Yuqicheng Zhu, Jingcheng Wu and Hongkuan Zhou. The work was partially supported by EU Projects Graph Massivizer (GA 101093202), enRichMyData (GA 101070284), SMARTY (GA 101140087), the EPSRC project OntoEm (EP/Y017706/1) and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - SFB 1574 - Project number 471687386. The authors also gratefully acknowledge the computing time provided on the high-performance computer HoreKa by the National High-Performance Computing Center at KIT (NHR@KIT). This center is jointly supported by the Federal Ministry of Education and Research and the Ministry of Science, Research and the Arts of Baden-Württemberg, as part of the National High-Performance Computing (NHR) joint funding program (https://www.nhr-verein.de/en/our-partners). HoreKa is partly funded by the German Research Foundation (DFG).

References

Anastasios Nikolas Angelopoulos, Stephen Bates, Michael I. Jordan, and Jitendra Malik. 2021a. Uncertainty sets for image classifiers using conformal prediction. In *ICLR*. OpenReview.net.

Anastasios Nikolas Angelopoulos, Stephen Bates, Michael I. Jordan, and Jitendra Malik. 2021b. Uncertainty sets for image classifiers using conformal prediction. In *ICLR*. OpenReview.net.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Margarida Campos, António Farinhas, Chrysoula Zerva, Mário AT Figueiredo, and André FT Martins. 2024. Conformal prediction for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 12:1497–1516.

Xuelu Chen, Michael Boratko, Muhao Chen, Shib Sankar Dasgupta, Xiang Lorraine Li, and Andrew Mccallum. 2021a. Probabilistic box embeddings for uncertain knowledge graph reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 882–893.

Xuelu Chen, Muhao Chen, Weijia Shi, Yizhou Sun, and Carlo Zaniolo. 2019. Embedding uncertain knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3363–3370.

Zhu-Mu Chen, Mi-Yen Yeh, and Tei-Wei Kuo. 2021b. Passleaf: a pool-based semi-supervised learning framework for uncertain knowledge graph embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4019–4026.

Ronald A Fisher. 1935. The fiducial argument in statistical inference. *Annals of eugenics*, 6(4):391–398.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37.

Kexin Huang, Ying Jin, Emmanuel Candes, and Jure Leskovec. 2024. Uncertainty quantification over graph with conformalized graph neural networks. *Advances in Neural Information Processing Systems*, 36.

Ramneet Kaur, Susmit Jha, Anirban Roy, Sangdon Park, Edgar Dobriban, Oleg Sokolsky, and Insup Lee. 2022. idecode: In-distribution equivariance for conformal out-of-distribution detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 7104–7114.

Angelika Kimmig, Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2012. A short introduction to probabilistic soft logic. In *Neural Information Processing Systems*.

Roger Koenker and Gilbert Bassett Jr. 1978. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. 2018. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111.

Lysimachos Maltoudoglou, Andreas Paisios, and Harris Papadopoulos. 2020. Bert-based conformal predictor for sentiment analysis. In *Conformal and Probabilistic Prediction and Applications*, pages 269–284. PMLR.

Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. 2018a. Never-ending learning. *Communications of the ACM*, 61(5):103–115.

Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha P. Talukdar, Bo Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matt Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil A. Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. 2018b. Never-ending learning. *Commun. ACM*, 61(5):103–115.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *ICML*, pages 809–816. Omnipress.

OpenAI. 2024. Chatgpt(3.5)[large language model]. https://chat.openai.com.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. 2002. Inductive confidence machines for regression. In *Machine learning: ECML 2002: 13th European conference on machine learning Helsinki, Finland, August 19–23, 2002 proceedings 13*, pages 345–356. Springer.

Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. 2013. Knowledge graph identification. In *The Semantic Web–ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I 12*, pages 542–557. Springer.

Ylli Sadikaj, Hongkuan Zhou, Lavdim Halilaj, Stefan Schmid, Steffen Staab, and Claudia Plant. 2025. Multiads: Defect-aware supervision for multi-type anomaly detection and segmentation in zero-shot learning. *CoRR*, abs/2504.06740.

Tara Safavi, Danai Koutra, and Edgar Meij. 2020. Evaluating the calibration of knowledge graph embeddings for trustworthy link prediction. In *EMNLP (1)*, pages 8308–8321. Association for Computational Linguistics.

Glenn Shafer and Vladimir Vovk. 2008. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3).

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017a. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017b. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Damian Szklarczyk, John H. Morris, Helen Cook, Michael Kuhn, Stefan Wyder, Milan Simonovic, Alberto Santos, Nadezhda T. Doncheva, Alexander Roth, Peer Bork, Lars Juhl Jensen, and Christian von Mering. 2017. The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible. *Nucleic Acids Res.*, 45(Database-Issue):D362–D368.

Damian Szklarczyk, John H Morris, Helen Cook, Michael Kuhn, Stefan Wyder, Milan Simonovic, Alberto Santos, Nadezhda T Doncheva, Alexander Roth, Peer Bork, et al. 2016. The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic acids research*, page gkw937.

Pedro Tabacof and Luca Costabello. 2020. Probability calibration for knowledge graph embedding models. In *ICLR*. OpenReview.net.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. 2005. *Algorithmic learning in a random world*, volume 29. Springer.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD international conference on management of data*, pages 481–492.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015a. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR (Poster)*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015b. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Soroush H Zargarbashi, Simone Antonelli, and Aleksandar Bojchevski. 2023. Conformal prediction sets for graph neural networks. In *International Conference on Machine Learning*, pages 12292–12318. PMLR.

Soroush H Zargarbashi and Aleksandar Bojchevski. 2023. Conformal inductive graph neural networks. In *The Twelfth International Conference on Learning Representations*.

Hongkuan Zhou, Lavdim Halilaj, Sebastian Monka, Stefan Schmid, Yuqicheng Zhu, Bo Xiong, and Steffen Staab. 2024. Visual representation learning guided by multi-modal prior knowledge. *CoRR*, abs/2410.15981.

Hongkuan Zhou, Stefan Schimid, Yicong Li, Lavdim Halilaj, Xiangtong Yao, and Wei Cao. 2025. Predicting the road ahead: A knowledge graph based foundation model for scene understanding in autonomous driving. In *The Semantic Web*, pages 116–132, Cham. Springer Nature Switzerland.

Yuqicheng Zhu, Daniel Hernández, Yuan He, Zifeng Ding, Bo Xiong, Evgeny Kharlamov, and Steffen Staab. 2025a. Predicate-conditional conformalized answer sets for knowledge graph embeddings. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4145–4167, Vienna, Austria. Association for Computational Linguistics.

Yuqicheng Zhu, Nico Potyka, Mojtaba Nayyeri, Bo Xiong, Yunjie He, Evgeny Kharlamov, and Steffen Staab. 2024a. Predictive multiplicity of knowledge graph embeddings in link prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 334–354.

Yuqicheng Zhu, Nico Potyka, Jiarong Pan, Bo Xiong, Yunjie He, Evgeny Kharlamov, and Steffen Staab. 2025b. Conformalized answer set prediction for knowledge graph embedding. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 731–750, Albuquerque, New Mexico. Association for Computational Linguistics.

Yuqicheng Zhu, Nico Potyka, Bo Xiong, Trung-Kien Tran, Mojtaba Nayyeri, Evgeny Kharlamov, and Steffen Staab. 2024b. Approximating probabilistic inference in statistical EL with knowledge graph embeddings. *CoRR*, abs/2407.11821.

Yuqicheng Zhu, Nico Potyka, Bo Xiong, Trung-Kien Tran, Mojtaba Nayyeri, Steffen Staab, and Evgeny Kharlamov. 2023. Towards statistical reasoning with ontology embeddings. In *ISWC (Posters/Demos/Industry)*, volume 3632 of *CEUR Workshop Proceedings*. CEUR-WS.org.

A Proof

Proposition 1. *Assume weighted triples in $\mathcal{T}$ and the test weighted triple (q_{n+1}, c_{n+1}) are i.i.d. For any confidence level $\alpha \in [0,1]$ and any nonconformity measure S, the ICP-based conformal predictor $\Gamma^{\alpha}_{\mathrm{ICP}}$ is conservatively valid:*

$$\mathbb{P}\big(c_{n+1} \in \Gamma^{\alpha}_{\mathrm{ICP}}(\mathcal{T}, q_{n+1})\big) \geq \alpha. \tag{23}$$

Furthermore, if $\{s_i\}_{i=m+1}^{n}$ contains no ties, $\Gamma^{\alpha}_{\mathrm{ICP}}$ is also asymptotically exactly valid:

$$\lim_{\ell \to \infty} \mathbb{P}\big(c_{n+1} \in \Gamma^{\alpha}_{\mathrm{ICP}}(\mathcal{T}, q_{n+1})\big) = \alpha. \tag{24}$$

Proof of the lower bound. Recall that the set of weighted triples $\mathcal{T}$ is partitioned into a proper training set $\mathcal{T}_{\mathrm{train}} = \{(q_i, c_i)\}_{i=1}^{m}$ and a calibration set $\mathcal{T}_{\mathrm{cal}} = \{(q_i, c_i)\}_{i=m+1}^{n}$ of size $\ell = n - m$.

By assuming all weighted triples

$$(q_{m+1}, c_{m+1}), \ldots, (q_{n+1}, c_{n+1}) \tag{25}$$

are i.i.d, we know that their order is a uniform random permutation of the indices $m+1, \ldots, n+1$. Hence the corresponding nonconformity scores $\{s_{m+1}, \ldots, s_{n+1}\}$ are exchangeable: every one of the $(\ell+1)!$ permutations of these scores is equally likely (Papadopoulos et al., 2002, Section 3), (Vovk et al., 2005, Chapter 4.2.2). Formally, for any permutation $\pi : \{m+1, \ldots, n+1\} \to \{m+1, \ldots, n+1\}$,

$$(s_{m+1}, \ldots, s_{n+1}) \overset{d}{=} (s_{\pi(m+1)}, \ldots, s_{\pi(n+1)}), \tag{26}$$

where $\overset{d}{=}$ denotes equality in distribution.

The ICP-based prediction interval at confidence level α includes a candidate score if and only if s_{n+1} is among the $\lceil \alpha(\ell+1) \rceil$ smallest s_i:

$$\frac{|\{i = m+1, \ldots, n+1 : s_i \geq s_{n+1}\}|}{\ell+1} > 1 - \alpha. \tag{27}$$

Due to the exchangeability in Equation (26), each of the $\ell+1$ positions that s_{n+1} could occupy among the scores $\{s_{m+1}, \ldots, s_{n+1}\}$ is equally likely. Thus, the probability that the prediction interval covers the ground truth scores equals

$$\mathbb{P}\big(c_{n+1} \in \Gamma^{\alpha}_{\mathrm{ICP}}(\mathcal{T}, q_{n+1})\big) = \frac{\lceil \alpha(\ell+1) \rceil}{\ell+1} \geq \alpha, \tag{28}$$

which establishes conservative coverage at level α. □

Proof of the upper bound. We prove the upper bound based on Lei et al. (2018, Appendix A.1). By assuming no ties in the set of nonconformity scores in $\{s_{m+1}, \ldots, s_{n+1}\}$, we know that the nonconformity scores in $\{s_{m+1}, \ldots, s_{n+1}\}$ are all distinct with probability one. Define $\alpha' = \alpha + 1/(\ell+1)$. Consider now the complementary set $\Gamma^{\alpha'}_{\mathrm{ICP}}$:

$$\Gamma^{\alpha'}_{\mathrm{ICP}}(\mathcal{T}, q_{n+1}) := \left\{ c \in [0,1] : \frac{|i = m+1, \ldots, n+1 : s_i \geq s_{n+1}|}{\ell+1} \leq 1 - \alpha' \right\}, \tag{29}$$

where

$$\begin{aligned} s_i &:= S\big(\mathcal{T}_{\mathrm{train}}, (q_i, c_i)\big), i = m+1, \ldots, n, \\ s_{n+1} &:= S\big(\mathcal{T}_{\mathrm{train}}, (q_{n+1}, c)\big). \end{aligned} \tag{30}$$

Due to the i.i.d. assumption and hence exchangeability of the nonconformity scores $\{s_{m+1}, \ldots, s_{n+1}\}$, the rank of s_{n+1} among these $\ell+1$ scores is uniformly distributed. Therefore, for any fixed $c \in [0,1]$,

$$\mathbb{P}\big(c_{n+1} \in \Gamma^{\alpha'}_{\mathrm{ICP}}(\mathcal{T}, q_{n+1})\big) = \frac{\lceil (1-\alpha')(\ell+1) \rceil}{\ell+1} \geq 1 - \alpha' = 1 - \alpha - \frac{1}{\ell+1} \tag{31}$$

Moreover, since we assumed no ties, the sets $\Gamma^{\alpha}_{\text{ICP}}(\mathcal{T}, q_{n+1})$ and $\Gamma^{\alpha'}_{\text{ICP}}(\mathcal{T}, q_{n+1})$ are disjoint:

$$\Gamma^{\alpha}_{\text{ICP}}(\mathcal{T}, q_{n+1}) \cap \Gamma^{\alpha'}_{\text{ICP}}(\mathcal{T}, q_{n+1}) = \emptyset \tag{32}$$

Thus,

$$\begin{aligned}
&\mathbb{P}\big(c_{n+1} \in \Gamma^{\alpha}_{\text{ICP}}(\mathcal{T}, q_{n+1})\big) + \mathbb{P}\big(c_{n+1} \in \Gamma^{\alpha'}_{\text{ICP}}(\mathcal{T}, q_{n+1})\big) \leq 1 \\
\Rightarrow &\mathbb{P}\big(c_{n+1} \in \Gamma^{\alpha}_{\text{ICP}}(\mathcal{T}, q_{n+1})\big) \leq 1 - \mathbb{P}\big(c_{n+1} \in \Gamma^{\alpha'}_{\text{ICP}}(\mathcal{T}, q_{n+1})\big) \\
\Rightarrow &\mathbb{P}\big(c_{n+1} \in \Gamma^{\alpha}_{\text{ICP}}(\mathcal{T}, q_{n+1})\big) \leq \alpha + \frac{1}{\ell + 1}
\end{aligned}$$

Combine this upper bound with the lower bound proved earlier; together they give

$$\alpha \leq \mathbb{P}\big(c_{n+1} \in \Gamma^{\alpha}_{\text{ICP}}(\mathcal{T}, q_{n+1})\big) \leq \alpha + \frac{1}{\ell + 1} \tag{33}$$

Hence $\mathbb{P}\big(c_{n+1} \in \Gamma^{\alpha}_{\text{ICP}}(\mathcal{T}, q_{n+1})\big) \xrightarrow[\ell \to \infty]{} \alpha$ establishing *exact* asymptotic validity when ties occur with probability 0. □

B Details of Experimental Settings

B.1 Details of Datasets

We provide further details on the three benchmark datasets used in our experiments: CN15k, NL27k, and PPI5k.

CN15k is derived from ConceptNet (Speer et al., 2017b), a multilingual commonsense knowledge graph. Each assertion has a confidence score between 0.1 and 22, with 99.6% of scores below or equal to 3.0. Following Chen et al. (2019), we cap the scores at 3.0 and then apply log by min-max normalization to map the values into the range $[0.5, 1.0]$.

NL27k is built from NELL (Mitchell et al., 2018b), a large-scale English knowledge base constructed via semi-automatic extraction. Confidence scores are assigned based on iterative self-training and rule-based extraction pipelines. These scores, originally in $[0.1, 0.9]$, are min-max normalized to $[0.1, 1.0]$.

PPI5k is derived from STRING (Szklarczyk et al., 2017), a biological database of protein-protein interactions. Each triple represents an interaction between two proteins, annotated with a confidence score between 0 and 1. These confidence scores can be interpreted as probabilities. For example, a score of 0.5 indicates that about half of the predicted interactions may be false positives. Higher scores suggest higher probability for a true biological interaction.

Data Splits. Following Chen et al. (2019), all datasets are partitioned into 85% for training, 7% for calibration, and 8% for testing. The data statistics and splits are shown in Table 4. These are used consistently across all models and experiments.

Dataset	#Entity	#Predicate	#Training/Calibration/Test Facts
CN15k	15,000	36	204,984/16,881/19,293
NL27k	27,221	404	149,100/12,278/14,034
PPI5k	4,999	7	230,929/19,017/21,720

Table 4: Dataset statistics used in our experiments.

B.2 Details of UnKGE Backbones

UKGE (Chen et al., 2019) extends KGE methods by explicitly modeling uncertainty through confidence scores with each triple. It adapts the scoring function and loss function to predict continuous values in $[0, 1]$ that reflect the plausibility of triples. Given a triple $\langle h, r, t\rangle$, and following the DistMult model (Yang et al., 2015b), the UKGE score function is defined as:

$$f\left((\mathbf{h} \circ \mathbf{t})^\top \mathbf{r}\right) \tag{34}$$

where $\mathbf{h}$, $\mathbf{r}$, and $\mathbf{t}$ denote the embeddings of the head entity, relation, and tail entity, respectively and $f : \mathbb{R} \to [0, 1]$ is a normalization function that maps the raw score to a confidence score in $[0, 1]$.

UKGE offers two variants, $\text{UKGE}_{\text{logi}}$ and $\text{UKGE}_{\text{rect}}$, which differ in how they map raw triple scores to confidence scores. We primarily focus on $\text{UKGE}_{\text{logi}}$, which applies a learnable logistic function to map the raw triple score to a confidence score. Specifically, given a triple $\langle h, r, t\rangle$ with embeddings $\mathbf{h}$, $\mathbf{r}$, and $\mathbf{t}$, the score is computed as:

$$f_{\text{logi}}(h, r, t) = \frac{1}{1 + \exp\left(-\left(w(\mathbf{h} \circ \mathbf{t})^\top \mathbf{r} + b\right)\right)}, \tag{35}$$

where w and b are learnable scalar parameters of the logistic function.

An alternative variant, $\text{UKGE}_{\text{rect}}$, adopts a rectified and bounded linear transformation:

$$f_{\text{rect}}(h, r, t) = \min\left(\max\left(w(\mathbf{h} \circ \mathbf{t})^\top \mathbf{r} + b, 0\right), 1\right). \tag{36}$$

Probabilistic Soft Logic (PSL) (Kimmig et al., 2012) is used to estimate confidence scores c for unseen weighted triples, resulting in an extended weighted triple set $\mathcal{T}^{\text{psl}}$ that augments the original training dataset. We define the augmented training set as $\mathcal{T}^+ = \mathcal{T} \cup \mathcal{T}^{\text{psl}}$ and let $\mathcal{T}^-$ denote the set f negative weighted triples. The loss is then defined as:

$$L = \sum_{(q,c)\in\mathcal{T}^+} (M_\theta(q) - c)^2 + \alpha \sum_{(q,c)\in\mathcal{T}^-} M_\theta(q)^2, \tag{37}$$

Note that c is 0 for weighted triples in $\mathcal{T}^-$, and $\alpha \in \mathbb{R}^+$ is a hyperparameter controlling the penalty on unobserved triples in $\mathcal{T}^-$.

PASSLEAF (Chen et al., 2021b) extends the UKGE framework by incorporating a semi-supervised learning strategy that leverages pseudo-labeled triples to better utilize uncertain information. Its training objective consists of three components: a supervised loss over observed positive triples, a loss over generated negative triples, and a semi-supervised loss over pseudo-labeled triples.

The supervised loss L_{pos} minimizes the mean squared error between the model's predicted confidence scores and the ground-truth scores c for each

weighted triple $(q, c) \in \mathcal{T}$:

$$L_{\text{pos}} = \sum_{(q,c)\in\mathcal{T}} |M_\theta(q) - c|^2 . \tag{38}$$

The negative sample loss L_{neg} encourages the model to assign low confidence scores to generated negative triples $(q, c) \in \mathcal{T}^-$:

$$L_{\text{neg}} = \sum_{(q,c)\in\mathcal{T}^-} |M_\theta(q)|^2 . \tag{39}$$

PASSLEAF introduces an additional semi-supervised loss L_{semi} over a pseudo-labeled set $\mathcal{T}^{\text{semi}}$, where each query triple is assigned a confidence score based on the model's own predictions from a prior training stage.

$$L_{\text{semi}} = \sum_{(q,c)\in\mathcal{T}^{\text{semi}}} |M_\theta(q) - c|^2 . \tag{40}$$

The overall objective combines all components, with the semi-supervised and negative losses normalized by the total number of generated triples:

$$L = L_{\text{pos}} + \frac{1}{|\mathcal{T}^{\text{semi}} \cup \mathcal{T}^-|} \left(L_{\text{semi}} + L_{\text{neg}}\right). \tag{41}$$

BEUrRE (Chen et al., 2021a) models entities and relations as probabilistic boxes in a latent space. This approach supports detailed modeling of uncertainty at both the fact and entity levels. Given a triple $q = \langle h, r, t\rangle$, the model defines the confidence score $M_\theta(q)$ as an approximate conditional probability:

$$M_\theta(q) = \frac{\mathbb{E}\left[\text{Vol}\left(H_r(Box_h) \cap T_r(Box_t)\right)\right]}{\mathbb{E}\left[\text{Vol}\left(T_r(Box_t)\right)\right]}, \tag{42}$$

where Box_h and Box_t denote the probabilistic boxes corresponding to entities h and t, respectively. The functions H_r and T_r are relation-specific transformations, typically implemented as affine mappings. $\text{Vol}(\cdot)$ denotes the volume of a box, and $\mathbb{E}[\cdot]$ represents the expectation taken over the stochastic parameters of the boxes. BEUrRE is trained using the same loss function as defined in Equation (37).

B.3 Details of Confidence Predictor

Fisher Prediction Intervals (FPI) (Fisher, 1935) provide a classical statistical approach to quantifying uncertainty in predicted confidence scores. FPI relies on assumptions: (i) the residuals (i.e. additive noise of the regression model) are approximately normally distributed, (ii) the calibration examples are independent and identically distributed (i.i.d.), and (iii) the prediction variance is constant across instances (homoscedasticity). Under these assumptions, FPI offers a parametric interval estimation framework using the t-distribution. Given a *calibration set* $\mathcal{T}_{\text{cal}} = \{tr_i\}_{i=m+1}^{n}$ of size $\ell = n - m$, consisting of predicted confidence scores $\{c_{m+1}, \ldots, c_n\}$, we compute the sample mean $\bar{c}_{\text{cal}}$ and unbiased sample variance s^2_{cal} as:

$$\bar{c}_{\text{cal}} = \frac{1}{\ell} \sum_{i=m+1}^{n} c_i, \tag{43}$$

$$s^2_{\text{cal}} = \frac{1}{\ell - 1} \sum_{i=m+1}^{n} (c_i - \bar{c}_{\text{cal}})^2. \tag{44}$$

The FPI-based prediction interval for a new query q_{n+1} is given by:

$$\Gamma^\alpha_{\text{FPI}}(\mathcal{T}, q_{n+1}) := \left[\bar{c}_{\text{cal}} - t^{(1-\alpha)/2}_{\ell-1} \cdot s_{\text{cal}} \sqrt{\frac{\ell}{\ell-1}}, \right.$$
$$\left. \bar{c}_{\text{cal}} + t^{(1-\alpha)/2}_{\ell-1} \cdot s_{\text{cal}} \sqrt{\frac{\ell}{\ell-1}}\right], \tag{45}$$

where $t^{(1-\alpha)/2}_{\ell-1}$ is the $(1-\alpha)/2$ quantile of the t-distribution with $\ell - 1$ degrees of freedom. For instance, setting $\alpha = 0.9$ yields a prediction interval with 90% confidence.

Quantile Regression (QR) (Koenker and Bassett Jr, 1978) provides a flexible approach to modeling the conditional distribution of predicted confidence scores without assuming any specific parametric form. Given the *proper training set* $\mathcal{T}_{\text{train}} = \{tr_i\}_{i=1}^{m}$, we train two separate quantile regressors to construct prediction intervals: a lower quantile model at $\tau_{\text{lower}} = (1-\alpha)/2$ and an upper quantile model at $\tau_{\text{upper}} = 1 - (1-\alpha)/2$, where α denotes the desired confidence level.

Each quantile model M_θ is optimized by minimizing the pinball loss function:

$$L = \sum_{(q,c)\in\mathcal{T}_{\text{train}}} \tau \cdot \max\left(c - M_\theta(q), 0\right) + (1-\tau) \cdot \max\left(M_\theta(q) - c, 0\right), \tag{46}$$

The resulting prediction interval for a new query q_{n+1} is obtained by evaluating the two trained models with confidence level α:

$$\Gamma^\alpha_{\text{QR}}(\mathcal{T}, q_{n+1}) := \left[\hat{M}_\theta^{\text{lower}}(q_{n+1}), \hat{M}_\theta^{\text{upper}}(q_{n+1})\right], \tag{47}$$

where $\hat{M}_\theta^{\text{lower}}$ and $\hat{M}_\theta^{\text{upper}}$ denote the lower and upper quantile predictors, respectively. QR enables instance-dependent prediction intervals that adapt to heteroscedasticity and local uncertainty patterns.

B.4 Implementation Details

All experiments are conducted on a single NVIDIA A100 Tensor Core GPU. Each experiment is repeated with 10 different global random seeds to ensure full reproducibility, including model initialization and data shuffling.

Hyperparameters for each model–dataset configuration are independently tuned via grid search. We search over learning rates $\{0.0001, 0.001, 0.01\}$, embedding dimensions $\{64, 128, 256, 512\}$, and batch sizes $\{128, 256, 512, 1024, 2048, 4096\}$.

Both variants of UKGE and PASSLEAF use early stopping based on the mean of validation loss and negative-sample validation loss, with patience set to 200 epochs due to slower convergence. BEUrRE, which converges more quickly, uses a shorter patience of 50 epochs. These values were empirically determined based on validation performance. Negative sampling ratios follow prior work: UKGE and PASSLEAF use 10 negative samples per positive triple, while BEUrRE, which benefits from higher negative pressure due to its probabilistic box structure, uses 30 negatives per positive.

The final hyperparameters, selected using validation loss, are as follows: for UKGE, learning rate = 0.001, embedding dimension = 128, and batch size = 128 for CN15k and NL27k, or 256 for PPI5k. PASSLEAF consistently uses learning rate = 0.001, embedding dimension = 512, batch size = 512, and the Adam optimizer, with additional semi-supervised settings: $T_{\text{NEW\_SEMI}} = 20$, $T_{\text{SEMI\_TRAIN}} = 30$, $M_{\text{SEMI}} = 0.8\times$ batch size, sample pool size $C = 10^7$, and $\alpha = 0.02$. BEUrRE uses a learning rate of 0.0001, embedding dimension = 64, $\beta = 0.01$, and batch size = 4096 for CN15k, or 2048 for NL27k and PPI5k.

C Complete Results

For completeness, we provide the full set of experimental results, including those omitted from the main paper due to space constraints. This includes detailed coverage and sharpness values across all datasets, UnKGE models, and baselines (Figures 4–5), as well as corresponding results on negative test triples (Figures 6–8). We also include comprehensive results for the conditionality analysis (Figures 9–13) and the sample efficiency analysis of the calibration step (Figures 14–18). These results further support and strengthen the empirical findings presented in Section 5.

D AI Assistants In Writing

We use ChatGPT (OpenAI, 2024) to enhance our writing skills, abstaining from its use in research and coding endeavors.

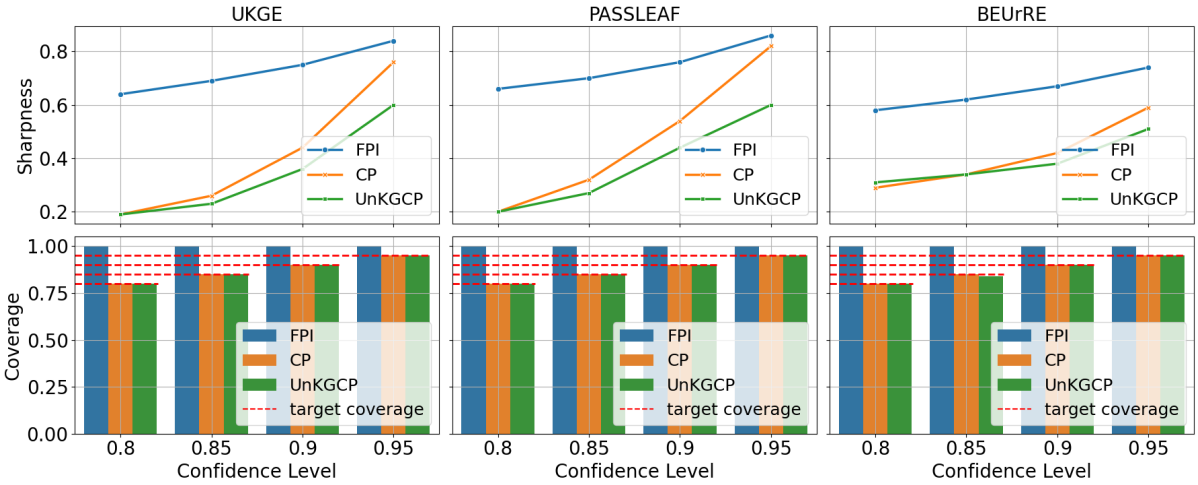

Figure 4: Effect of the confidence level α on the sharpness (top) and coverage (bottom) for **positive** test triples on NL27k. Each curve represents one predictor. Red dashed lines indicate the desired coverage levels.

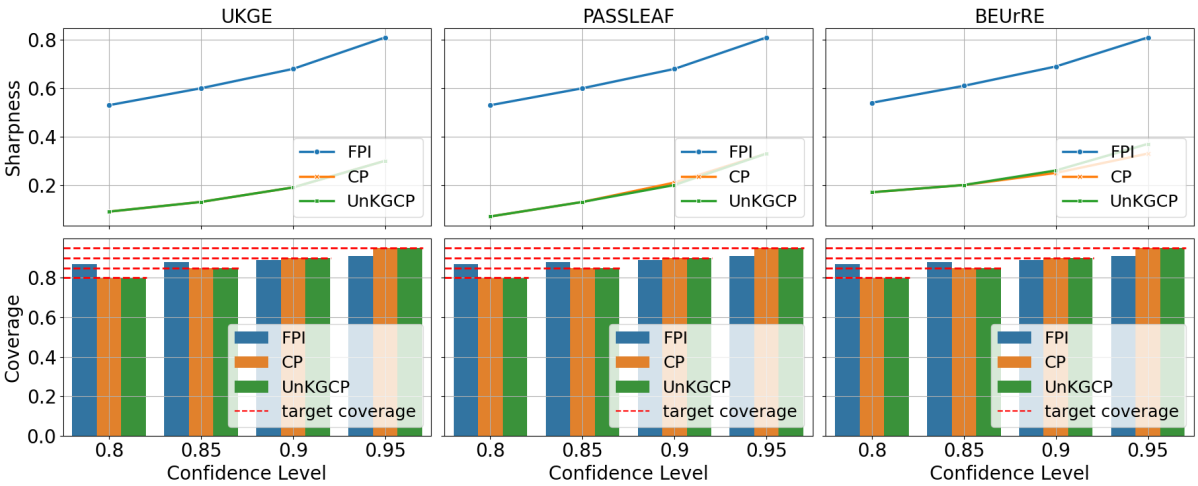

Figure 5: Effect of the confidence level α on the sharpness (top) and coverage (bottom) for **positive** test triples on PPI5k. Each curve represents one predictor. Red dashed lines indicate the desired coverage levels.

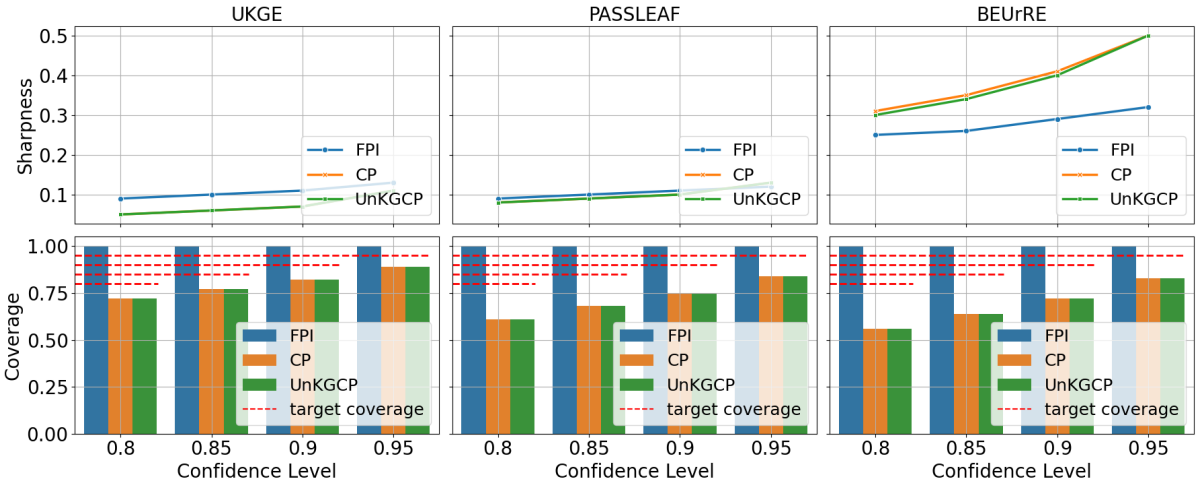

Figure 6: Effect of the confidence level α on the sharpness (top) and coverage (bottom) for **negative** test triples on CN15k. Each curve represents one predictor. Red dashed lines indicate the desired coverage levels.

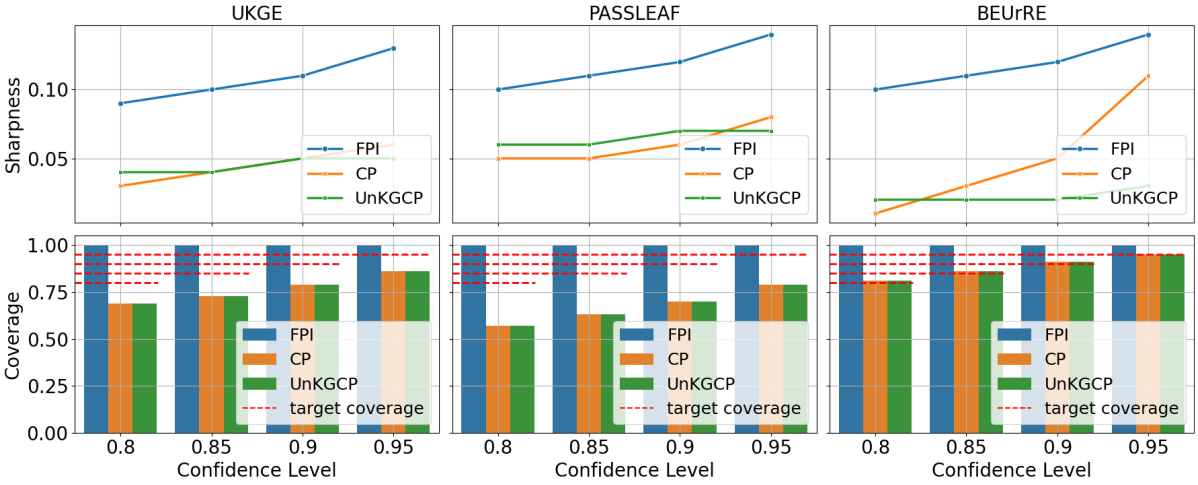

Figure 7: Effect of the confidence level α on the sharpness (top) and coverage (bottom) for **negative** test triples on NL27k. Each curve represents one predictor. Red dashed lines indicate the desired coverage levels.

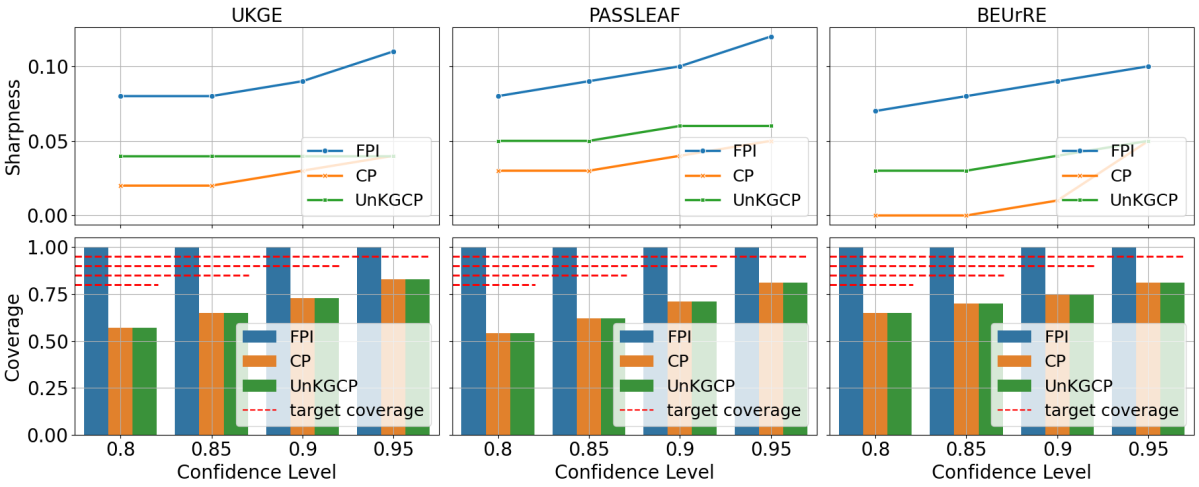

Figure 8: Effect of the confidence level α on the sharpness (top) and coverage (bottom) for **negative** test triples on PPI5k. Each curve represents one predictor. Red dashed lines indicate the desired coverage levels.

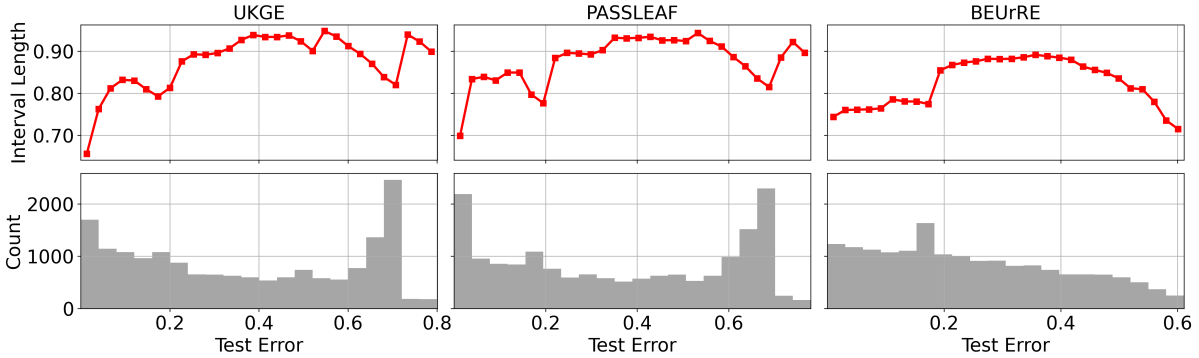

Figure 9: Conditionality analysis on CN15k (positive examples).

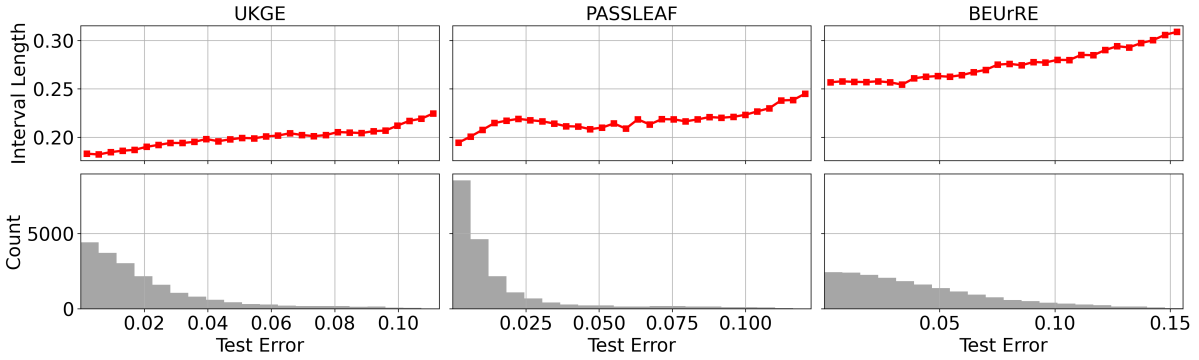

Figure 10: Conditionality analysis on PPI5k (positive examples).

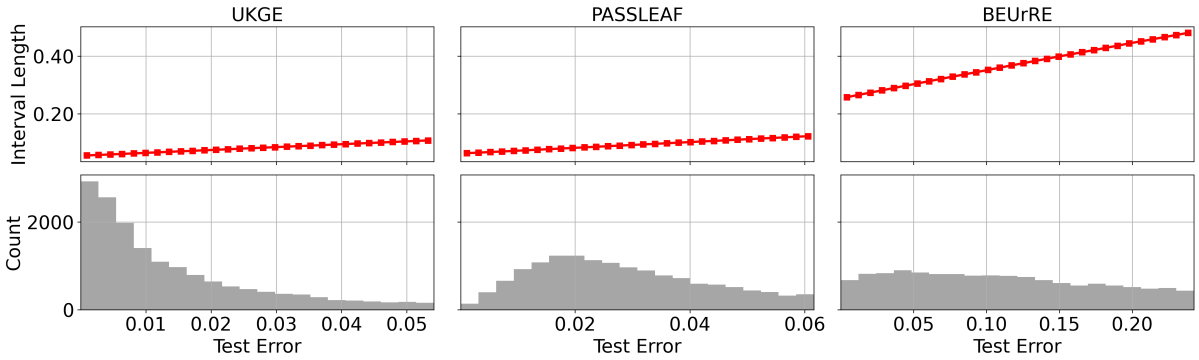

Figure 11: Conditionality analysis on CN15k (negative examples).

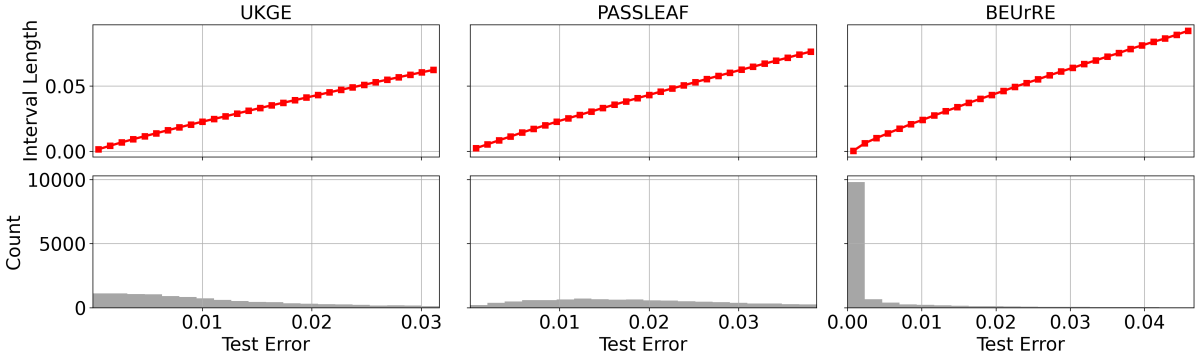

Figure 12: Conditionality analysis on NL27k (negative examples).

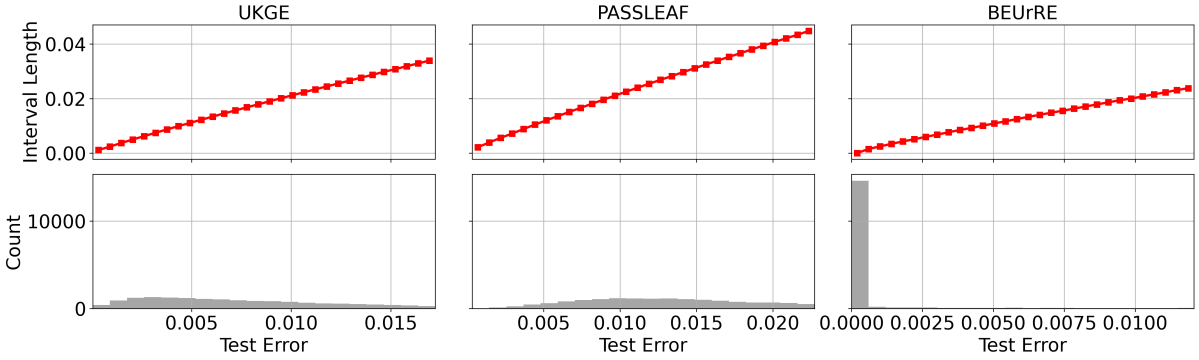

Figure 13: Conditionality analysis on PPI5k (negative examples).

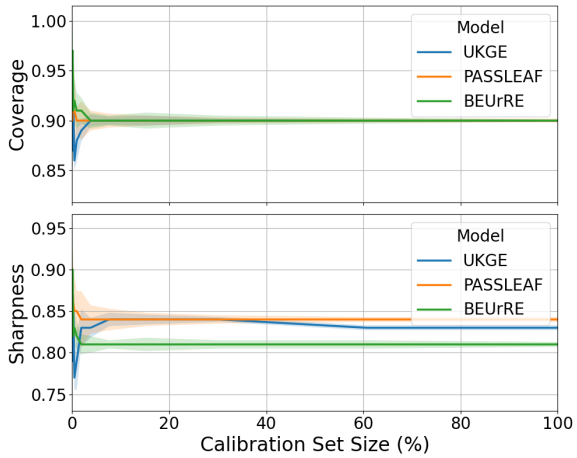

Figure 14: Effect of calibration set size on coverage and sharpness on CN15k (positive examples). The top panel reports coverage and the bottom panel reports sharpness. In both plots, the lines represent mean values across 10 runs, and the shaded areas indicate the standard deviation.

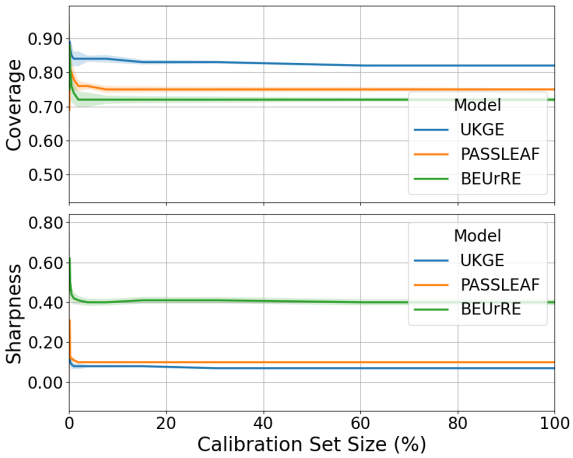

Figure 16: Effect of calibration set size on coverage and sharpness on CN15k (negative examples). The top panel reports coverage and the bottom panel reports sharpness. In both plots, the lines represent mean values across 10 runs, and the shaded areas indicate the standard deviation.

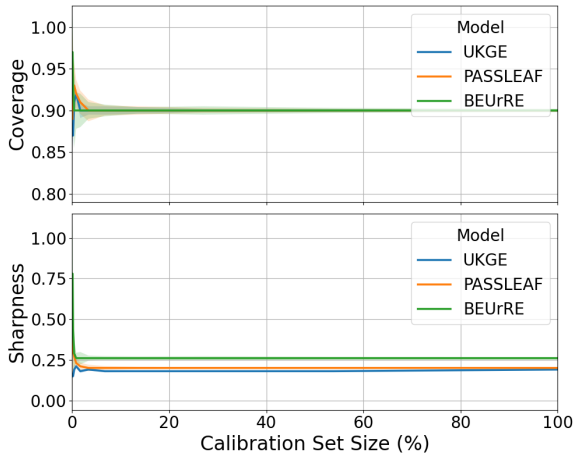

Figure 15: Effect of calibration set size on coverage and sharpness on PPI5k (positive examples). The top panel reports coverage and the bottom panel reports sharpness. In both plots, the lines represent mean values across 10 runs, and the shaded areas indicate the standard deviation.

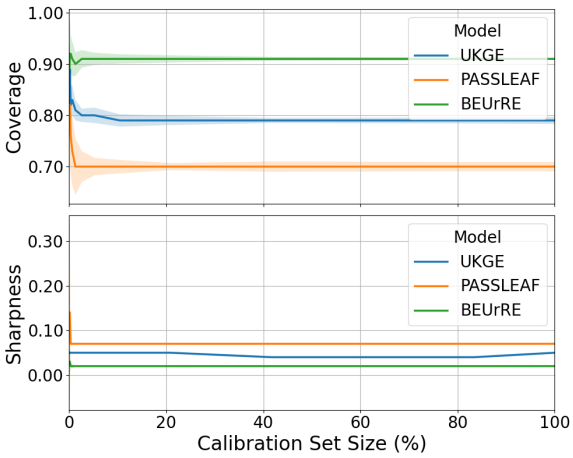

Figure 17: Effect of calibration set size on coverage and sharpness on NL27k (negative examples). The top panel reports coverage and the bottom panel reports sharpness. In both plots, the lines represent mean values across 10 runs, and the shaded areas indicate the standard deviation.

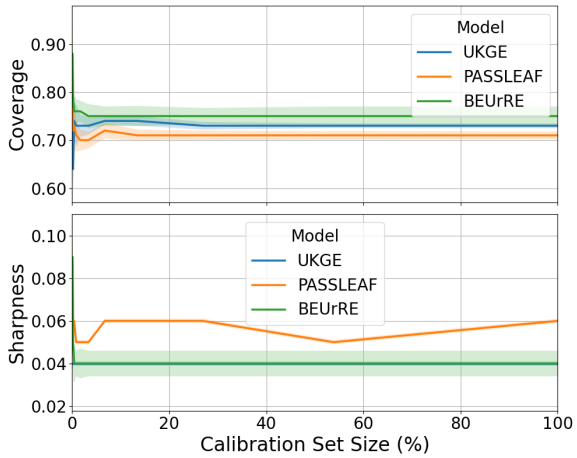

Figure 18: Effect of calibration set size on coverage and sharpness on PPI5k (negative examples). The top panel reports coverage and the bottom panel reports sharpness. In both plots, the lines represent mean values across 10 runs, and the shaded areas indicate the standard deviation.