Predicate-Conditional Conformalized Answer Sets for Knowledge Graph Embeddings

Yuqicheng Zhu[1,2], Daniel Hernández[1], Yuan He[3], Zifeng Ding[4], Bo Xiong[5], Evgeny Kharlamov[2,6], Steffen Staab[1,7]
[1]University of Stuttgart, [2]Bosch Center for AI,
[3]University of Oxford, [4]University of Cambridge, [5]Stanford University,
[6]University of Oslo, [7]University of Southampton
yuqicheng.zhu@de.bosch.com

Abstract

Uncertainty quantification in Knowledge Graph Embedding (KGE) methods is crucial for ensuring the reliability of downstream applications. A recent work applies conformal prediction to KGE methods, providing uncertainty estimates by generating a set of answers that is guaranteed to include the true answer with a predefined confidence level. However, existing methods provide probabilistic guarantees averaged over a reference set of queries and answers (*marginal coverage guarantee*). In high-stakes applications such as medical diagnosis, a stronger guarantee is often required: the predicted sets must provide consistent coverage per query (*conditional coverage guarantee*). We propose CondKGCP, a novel method that approximates predicate-conditional coverage guarantees while maintaining compact prediction sets. CondKGCP merges predicates with similar vector representations and augments calibration with rank information. We prove the theoretical guarantees and demonstrate empirical effectiveness of CondKGCP by comprehensive evaluations.

1 Introduction

Knowledge Graph Embeddings (KGE) encode entities and predicates as numerical vectors, enabling reasoning by exploiting similarities and analogies between entities and relations (Wang et al., 2017; Biswas et al., 2023). While KGE methods have demonstrated effectiveness in various downstream tasks such as link prediction (Bordes et al., 2013; Nickel et al., 2011) and question answering (Saxena et al., 2020), there remains uncertainty regarding the reliability of their predictions. Specifically, KGE models fail to identify when the answers to a query are uncertain (Zhu et al., 2024a).

Conformal prediction is a framework to quantify uncertainty by providing a *prediction set*—a set of possible solutions for a given task—that is guaranteed to cover the ground truth solution with a predefined confidence level (Vovk et al., 2005). By assigning a score to each possible solution, the method defines a threshold to choose the minimum number of elements for the prediction set to provide the coverage guarantee. Thus, the size of the predicted set reflects the uncertainty of the predictions, with larger sets indicating higher uncertainty.

Recently, Zhu et al. (2025) introduced a method, Conformalized Knowledge Graph Embedding (KGCP), which applies conformal prediction to quantify uncertainty in the predictions from KGE models. They show that KGCP provides *marginal coverage guarantees*, ensuring that the prediction sets meet the desired confidence level on average across all queries. However, predictive uncertainty may vary substantially across predicates, necessitating tailored coverage guarantees conditioned on predicates (*predicate-conditional coverage guarantees*). Such conditional guarantees are especially crucial for real-world applications where specific subgroups demand reliable uncertainty estimates. For instance, in a medical diagnosis system leveraging KGE, predicates like "contraindicated_for" (indicating that a treatment is not recommended for certain patients) and "has_symptom" (indicating that a specific disease or condition is associated with certain symptoms) may require different thresholds to achieve prediction sets with the desired confidence level. A shared threshold might fail to cover the true answer for the predicate "contraindicated_for", as it is often associated with fewer triples and demands a higher threshold.

Conditional coverage guarantee can be achieved by performing conformal prediction at the subgroup level (Vovk et al., 2005). However, the highly imbalanced distribution of triples across predicates in KGs (Xiong et al., 2018) poses challenges, often resulting in prediction sets that are either overly large or fail to cover the true answer (Ding et al., 2024; Shi et al., 2024). To address this limitation, we propose CondKGCP, a method de-

signed to approximate predicate-conditional guarantee while maintaining compact prediction sets. The key components of CondKGCP are as follows: (1) it **merges predicates** with similar vector representation to increase the number of calibration triples available for reliable subgroup-level conformal prediction, and (2) it introduces a **dual calibration schema** that combines score calibration with rank calibration to exclude noisy answer entities, thereby reducing the size of prediction sets.

We provide theoretical guarantees that CondKGCP achieves conditional coverage probabilities tightly centered around the desired confidence level and that the dual calibration schema reduces expected prediction set sizes under certain conditions. Empirically, we demonstrate that CondKGCP outperforms five baseline methods, achieving a superior trade-off between conditional coverage probability and prediction set size across commonly used benchmark datasets.

2 Related Work

The majority of KGE methods aim to improve model performance by capturing relational patterns through more expressive embedding spaces, such as complex (Trouillon et al., 2016; Sun et al., 2019), hyperbolic (Xiong et al., 2022), or probabilistic spaces (He et al., 2015). By enabling richer representations, these methods have shown strong performance across downstream tasks including query answering (Ren et al., 2020; He et al., 2024, 2025), recommendation (Sun et al., 2018), and image classification (Zhou et al., 2024). Despite these successes, uncertainty quantification in KGE remains largely underexplored. Most uncertainty quantification methods for KGE calibrate the plausibility scores generated by the models (Tabacof and Costabello, 2020; Safavi et al., 2020). However, these methods lack formal guarantees for the resulting probabilities. In contrast, Zhu et al. (2025) introduce an approach that provides formal statistical guarantees.

It is well-established that no prediction interval can achieve a conditional coverage guarantee in a finite sample without additional assumptions about the data distribution (Vovk, 2012; Foygel Barber et al., 2021). Consequently, many of the existing works provide coverage guarantees conditioned on specific subgroups, such as class-conditional coverage guarantees (Ding et al., 2024; Shi et al., 2024). Two main strategies have been proposed to improve conditional coverage probabilities. The first involves modifying the nonconformity measure. For instance, Romano et al. (2020) enhance conditional coverage by defining cumulative probability of ground truth as nonconformity score, though their approach often results in larger prediction sets. To reduce the size of prediction sets, Angelopoulos et al. (2021) introduce a regularization term in the nonconformity score. The second strategy leverages subgroup-level conformal prediction. Vovk et al. (2005) propose Mondrian Conformal Prediction, which performs conformal prediction within specific subgroups. Building on this, Ding et al. (2024) cluster calibration points based on the distribution of nonconformity scores, balancing the trade-off between conditional coverage probability and the size of the prediction sets. Additionally, Shi et al. (2024) further optimize prediction set sizes by incorporating rank information during the calibration step. However, these methods focus on classification, while our approach targets KGE-based link prediction, which is more challenging due to the large number of potential answers and the highly imbalanced triple distribution across predicates.

3 Preliminaries

3.1 Notations

Given two finite sets E and R whose elements are called *entities* and *predicates*, a *knowledge graph (KG)* is a subset of $E \times R \times E$, whose elements are known as *triples*. A *query* q is either an expression of the form $\langle h, r, ? \rangle$ or $\langle ?, r, t \rangle$, where $h, t \in E$, $r \in R$, and the question mark denotes the missing entity that we need to find. Given a query q, $\mathrm{pred}(q)$ is the predicate of the query, and $\mathrm{tr}(q, e)$ is the triple that results from assuming that e is an answer to the query. That is, $\mathrm{pred}(\langle ?, r, t \rangle) = r$, $\mathrm{pred}(\langle h, r, ? \rangle) = r$, $\mathrm{tr}(\langle ?, r, t \rangle, e) = \langle e, r, t \rangle$ and $\mathrm{tr}(\langle h, r, ? \rangle, e) = \langle h, r, e \rangle$.

A *query-answer* set $\mathcal{T}$ is a finite set of pairs (q, e) where q is a query, $e \in E$ is an answer to the query. Abusing notation, given a triple tr, we write $tr \in \mathcal{T}$ if there is a pair $(q, e) \in \mathcal{T}$ such that $\mathrm{tr}(q, e) = tr$. We use the names $\mathcal{T}_{\mathrm{tr}}$, $\mathcal{T}_{\mathrm{neg}}$, $\mathcal{T}_{\mathrm{cal}}$, and $\mathcal{T}_{\mathrm{test}}$ for the query-answer sets that are usually called *training set*, *negative triples set*, *calibration set*, and *test set*.

KGE methods train KGE models $M_\theta : E \times R \times E \to \mathbb{R}$ with parameters θ using a given training set $\mathcal{T}_{\mathrm{tr}}$ sampled from a distribution $\mathcal{P}$, whose elements are called *positive triples*, and a set $\mathcal{T}_{\mathrm{neg}}$ of negative

triples, disjoint with $\mathcal{T}_{\text{tr}}$. The learned model assigns scores to triples indicating their plausibility. It gives higher scores to the positive triples, and lower scores to the negative triples (Bordes et al., 2013; Nickel et al., 2011).

The performance of a KGE model is typically evaluated by the rank of answers in the test set. Given a pair $(q, e) \in \mathcal{T}_{\text{test}}$, the *rank* of answer e to query q predicted by M_θ, denoted $\text{rank}_{M_\theta}(q, e)$ is the size of the set of elements $E \ni e'$ such that $M_\theta(\text{tr}(q, e')) \geq M_\theta(\text{tr}(q, e))$. Smaller rank values indicate a better model performance.

3.2 Conformalized KGE

Given a KGE model M_θ trained on $\mathcal{T}_{\text{tr}}$, a pair $(q, e) \in \mathcal{T}_{\text{test}}$, and a user-specified error rate $\epsilon \in [0, 1]$, KGCP (Zhu et al., 2025) provides a set of entities, which is guaranteed to contain e with a probability of at least $1 - \epsilon$. In this section, we provide background on the method.

A *nonconformity measure* $S : E \times R \times E \to \mathbb{R}$ quantifies how unusual a triple is with respect to the training set. This measure is typically derived from a pre-trained KGE model, e.g., $S(tr) = -M_\theta(tr)$ (Zhu et al., 2025). Based on the nonconformity measure, the procedure of conformal prediction consists of two steps:

Calibration Step: given a number $\tau \in [0, 1]$ and a finite set $A \subseteq \mathbb{R}$, the τ-quantile of A, denoted $\text{quant}(\tau, A)$, is infimum of the set of elements $a \in A$ such that $|\{b \in A : b \leq a\}|/|A| \geq \tau$. Given a query-answer set $\mathcal{T}$, the empirical *quantile of nonconformity scores* is:

$$\hat{s}_\epsilon(\mathcal{T}) = \text{quant}\left(\frac{\lceil(|\mathcal{T}| + 1)(1 - \epsilon)\rceil}{|\mathcal{T}|}, \mathcal{T}\right). \quad (1)$$

Given the calibration set $\mathcal{T}_{\text{cal}}$, for a target coverage $1 - \epsilon$, we obtain the corresponding empirical quantile of nonconformity scores, $\hat{s}_\epsilon(\mathcal{T}_{\text{cal}})$.

Set Construction Step: Given a threshold s, and a query q, we define the set $E_q[S \leq s]$ as follows:

$$E_q[S \leq s] = \{e \in E : S(\text{tr}(q, e)) \leq s\}. \quad (2)$$

The *prediction set* for a test query q, denoted $\hat{C}(q)$, is then constructed by including all answer entities that have nonconformity scores smaller than the threshold $\hat{s}_\epsilon(\mathcal{T}_{\text{cal}})$:

$$\hat{C}(q) = E_q[S \leq \hat{s}_\epsilon(\mathcal{T}_{\text{cal}})]. \quad (3)$$

Theorem 1 (Zhu et al. (2025)). *Suppose the triples in $\mathcal{T}_{\text{tr}}$, $\mathcal{T}_{\text{cal}}$ and $\mathcal{T}_{\text{test}}$ are drawn independent and identically distributed (i.i.d) from the underlying distribution $\mathcal{P}$. For every element $(q, e) \in \mathcal{T}_{\text{test}}$, the probability of e to being included in the prediction set of q satisfies the following bounds: (i) $\mathbb{P}(e \in \hat{C}(q)) \geq 1 - \epsilon$, and (ii) if there is no tie in the set of scores of the triples in $\mathcal{T}_{\text{cal}}$, then $\mathbb{P}(e \in \hat{C}(q)) \leq 1 - \epsilon + \frac{1}{|\mathcal{T}_{\text{cal}}|+1}$.*

4 Conditional Conformal Prediction for Knowledge Graph Embedding (CondKGCP)

The goal of this paper is to approximate predicate-conditional coverage guarantee. Given a pair $(q, e) \in \mathcal{T}_{\text{test}}$, and an arbitrary predicate $r \in R$, Equation (4) defines the *predicate-conditional coverage guarantee*, which ensures that the true answer e is included in the prediction set of query q with a probability of at least $1 - \epsilon$.

$$\mathbb{P}(e \in \hat{C}(q) \mid \text{pred}(q) = r) \geq 1 - \epsilon \quad (4)$$

Equation (4) does not necessarily hold for KGCP because the predictive uncertainty and the nonconformity score distribution can vary dramatically across predicates, which violates the i.i.d assumption in Theorem 1.

To have the guarantee in Equation (4), a method called Mondrian Conformal Prediction (MCP) (Vovk et al., 2005) performs conformal prediction separately for each predicate. Given a subset $A \subseteq R$, let $\mathcal{T}_{\text{cal}}[A]$ be the query-answer subset of $\mathcal{T}_{\text{cal}}$ such that $tr \in \mathcal{T}_{\text{cal}}[A]$ if and only if $\text{pred}(tr) \in A$. The prediction set for the method MCP is defined as:

$$\hat{C}_{\text{MCP}}(q) = E_q[S \leq \hat{s}_\epsilon(\mathcal{T}_{\text{cal}}[\{r\}])]. \quad (5)$$

However, it is well known that most predicates in KGs are associated with very few triples (Xiong et al., 2018), resulting in small sets $\mathcal{T}_{\text{cal}}[\{r\}]$. This leads to unstable thresholds $\hat{s}_\epsilon(\mathcal{T}_{\text{cal}}[\{r\}])$, which in turn causes prediction sets to become overly large or fail to cover the ground truth.

To address these issues, we propose: (1) merging predicates to increase the number of triples in the calibration set, and (2) augmenting the calibration process with rank information to reduce the size of prediction sets.

4.1 Predicate Merging

To obtain a threshold $\hat{s}_\epsilon(\mathcal{T}_{\text{cal}}[\{r\}])$ that reliably covers the true answer with desired probability, it

is necessary to have a sufficiently large set $\mathcal{T}_{\text{cal}}[\{r\}]$ (Vovk et al., 2005; Ding et al., 2024). To increase $\mathcal{T}_{\text{cal}}[\{r\}]$, we merge predicates with highly similar vector representations. The rationale is that such predicates are likely to have similar distributions of nonconformity scores and, consequently, similar $\hat{s}_\epsilon(\mathcal{T}_{\text{cal}}[\{r\}])$. Formally, we aim to partition the set R such that each part corresponds to a subset of $\mathcal{T}_{\text{cal}}$ whose triples share similar predicates and is large enough to determine a reliable threshold for constructing prediction sets. To define a set partition, we propose Algorithm 1, which first places all predicates with enough data into separate partitions and then assigns predicates with few data to the partition of the most similar predicate.

Algorithm 1 Predicate Set Partition

Require: The set of predicates R, a natural number $\phi \leq \max_{r \in R} |\mathcal{T}_{\text{cal}}[\{r\}]|$, and a similarity function sim for pairs of predicates.

Ensure: A set partition P of R such that every part $A \in P$ satisfies $|\mathcal{T}_{\text{cal}}[A]| \geq \phi$.

$R_{\text{enough-data}} \leftarrow \{r \mid r \in R \text{ and } |\mathcal{T}_{\text{cal}}[\{r\}]| \geq \phi\}$

$R_{\text{few-data}} \leftarrow \{r \mid r \in R \text{ and } |\mathcal{T}_{\text{cal}}[\{r\}]| < \phi\}$

$\text{part}(r) \leftarrow \{r\}$, for each $r \in R_{\text{enough-data}}$

for $r' \in R_{\text{few-data}}$ **do**

  $r \leftarrow \text{argmax}_{r'' \in R_{\text{enough-data}}} \text{sim}(r', r'')$

  $\text{part}(r) \leftarrow \text{part}(r) \cup \{r'\}$

end for

$P \leftarrow \{\text{part}(r) \mid r \in R_{\text{enough-data}}\}$

Given a part g of the partition P defined by Algorithm 1, let $\mathcal{T}_g$ be the subset of $\mathcal{T}_{\text{cal}}$ consisting of all triples whose predicates belong to g.

In this work, we use negative Manhattan distance as the similarity measure. Let $(\boldsymbol{x})_i$ denote the i-th dimension of the vector $(\boldsymbol{x})_i$ representation of a predicate x, and d denote the number of dimensions. The similarity function is defined as

$$\text{sim}(a, b) = -\sum_{i=1}^{d} |(\boldsymbol{a})_i - (\boldsymbol{b})_i|. \tag{6}$$

4.2 Dual Calibration Schema

Prediction sets tend to be larger when conformal prediction is performed at the subgroup level due to the reduced number of calibration triples available for each subgroup compared to KGCP. To address this, and drawing inspiration from the recent work of Shi et al. (2024), we reduce the size of prediction sets by constructing prediction sets using a dual calibration schema that combines score calibration and rank calibration.

Given a query q with predicate belonging to g, the prediction set generated by CondKGCP is:

$$\hat{C}_{\text{CondKGCP}}(q) = \tag{7}$$
$$\Big\{e \in E_q[S \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g)] : \text{rank}_{M_\theta}(q, e) \leq \hat{k}(g)\Big\}.$$

It depends on two parameters, namely, the *score threshold* $\hat{s}_{\epsilon'(g)}(\mathcal{T}_g)$ and the *rank threshold* $\hat{k}(g)$, which we will define in the remainder of this section.

Rank Calibration. Recall that $\text{rank}_{M_\theta}(q, e)$ is the rank of answer e given query q. We define the *miscoverage error of top-k prediction set* for the part $g \in P$, denote ϵ_g^k, as follows:

$$\epsilon_g^k = \mathbb{P}(\text{rank}_{M_\theta}(q, e) > k \mid \text{pred}(q) \in g) \tag{8}$$

The rank threshold $\hat{k}(g)$ is selected such that $\epsilon_g^{\hat{k}(g)} < \epsilon$ to satisfy the coverage guarantee. However, achieving a smaller $\epsilon_g^{\hat{k}(g)}$ requires a larger $\hat{k}(g)$, which leads to larger prediction sets. To minimize the size of the prediction sets, we choose

$$\hat{k}(g) = \min\{k : \epsilon_g^k < \epsilon\}. \tag{9}$$

Score Calibration. We further apply a score threshold $\hat{s}_{\epsilon'(g)}(\mathcal{T}_g)$ for the entities that are ranked within top-$\hat{k}(g)$, where $\epsilon'(g) = \epsilon - \gamma\epsilon_g^{\hat{k}(g)}$ and γ is a hyperparameter.

Intuitively, the rank threshold $\hat{k}(g)$ filters out answer entities with large rank positions (high $\text{rank}_{M_\theta}(q, e)$), ensuring that CondKGCP performs score thresholding only on a subset of reliable test triples (Shi et al., 2024). The hyperparameter γ balances the trade-off between the conditional coverage guarantee and the size of prediction sets (see a detailed explanation in the next section).

5 Coverage & Size Guarantees

In this section, we will show the conditional coverage guarantee and size reduction guarantee of CondKGCP. All proofs are in Appendix A.

Proposition 1 (Conditional Coverage Guarantee)**.** *Let q be a query, e be its answer entity and p be the conditional coverage probability of* CondKGCP $p = \mathbb{P}(e \in \hat{C}_{\text{CondKGCP}}(q) \mid \text{pred}(q) \in g)$. *Given*

a user-specified error rate ϵ and a $\gamma \in [0,1]$, we have the following bounds for all parts $g \in P$:

$$p \geq 1 - \epsilon - (1-\gamma)\epsilon_g^{\hat{k}(g)}, \tag{10}$$

and if there is no tie in the set of nonconformity scores of the triples in $\mathcal{T}_g$, then

$$p \leq 1 - \epsilon + \gamma\epsilon_g^{\hat{k}(g)} + \frac{1}{|\mathcal{T}_g|+1}. \tag{11}$$

This proposition shows within each part g, the conditional coverage probability is close to $1-\epsilon$ with small controlled deviations. The deviation is governed by two "slack" terms: (1) the miscoverage error of rank calibration $\epsilon_g^{\hat{k}(g)}$ and (2) a finite-sample correction term $\frac{1}{|\mathcal{T}_g|+1}$ to handle ties. Both terms are very small, $\epsilon_g^{\hat{k}(g)}$ is guaranteed to be smaller than ϵ by the way we select $\hat{k}(g)$ in Section 4.2; $\frac{1}{|\mathcal{T}_g|+1}$ is also guaranteed to be smaller than $\frac{1}{\phi+1}$ since we make sure that every part has at least ϕ triples in Algorithm 1.

Note that γ does not affect the width of the coverage bounds but controls their asymmetry: a larger γ allows more deviation on the lower bound, while a smaller γ does so for the upper bound. Furthermore, γ influences the construction of the prediction sets by adjusting the score threshold via $\epsilon'(g)$: A larger γ reduces $\epsilon'(g)$, raising the threshold and yielding larger prediction sets, whereas a smaller γ results in smaller prediction sets. Thus, γ essentially deals with the trade-off between conditional coverage probability and the size of prediction sets.

Corollary 1 (Shi et al. (2024))**.** *Suppose $\epsilon'(g)$ and $\hat{k}(g)$ satisfy both following conditions*

$$\hat{k}(g) \in \left\{k : \epsilon_g^{\hat{k}(g)} < \epsilon\right\}; 0 \leq \epsilon'(g) \leq \epsilon - \epsilon_g^{\hat{k}(g)}, \tag{12}$$

the rank calibration guarantee to shrink the prediction sets, if for a query q and any $e' \in E$:

$$\begin{aligned}\mathbb{P}_q\Big(S(tr(q,e')) \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g), \\ \text{rank}_{M_\theta}(q,e') \leq \hat{k}(g)\Big) \\ \leq \mathbb{P}_q\Big(S(tr(q,e')) \leq \hat{s}_\epsilon(\mathcal{T}_g)\Big)\end{aligned} \tag{13}$$

Intuitively, the dual calibration schema tends to include less answer entities with high rank from KGE models thus reduce the size of prediction sets (Shi et al., 2024). The corollary demonstrate it is true in theory under the condition of Equation (13). We empirically verify the condition on benchmark datasets, the results in Appendix C show the practical utility of this corollary.

6 Experiment

We evaluate the CONDKGCP empirically and demonstrate its effectiveness in balancing the trade-off between predicate-conditional coverage probability and the size of the prediction sets.

6.1 Experimental Setup

Training KGE Models. We trained our KGE models using the LibKGE framework (Broscheit et al., 2020), following the hyperparameter search strategy described by Ruffinelli et al. (2019). All experiments were conducted on a Linux machine equipped with a 40GB NVIDIA A100 SXM4 GPU.

Datasets. We consider two widely-used benchmark datasets: WN18 and FB15k (Bordes et al., 2013). We follow Zhu et al. (2025, Appendix D.1) and do not consider their modified counterparts, WN18RR (Dettmers et al., 2018) and FB15k-237 (Toutanova and Chen, 2015) since they are not suitable for evaluating uncertainty quantification.

Baselines. We consider following methods as baselines: (1) KGCP (Zhu et al., 2025); (2) MCP (Vovk et al., 2005) which performs conformal prediction at the predicate-level; (3) CLUSTERCP (Ding et al., 2024), which clusters predicates based on similarity of score distribution and then conducts conformal prediction at the cluster level; (4) APS (Romano et al., 2020) and RAPS (Angelopoulos et al., 2021), which modify the nonconformity measure to provide improved conditional coverage probabilities (see details in Appendix E). Unless otherwise specified, we use the default nonconformity measure, SOFTMAX, as proposed by Zhu et al. (2025) (see the definition in Appendix D).

Hyperparameter Tuning. The validation set serves as the calibration set for all baselines. For CLUSTERCP, RAPS and CONDKGCP, we randomly sample triples from the training set (of the same sizes as the calibration set) to determine optimal hyperparameter settings. We follow hyperparameter search strategy from the original papers for CLUSTERCP and RAPS. For CONDKGCP, we tune $\gamma \in [0.01, 0.1, 0.5]$ and $\phi \in [20, 50, 100, 200]$. The best hyperparameters are reported in Table 7.

6.2 Evaluation Setup

We set the target coverage probability $1-\epsilon = 0.9$ by default, following Zhu et al. (2025). For each KGE method-dataset pair, we train the model 10 times, each time using a different random seed, and

Model	Methods	WN18 CovGap ↓	WN18 AveSize ↓	WN18 EF ↓	Model	Methods	FB15k CovGap ↓	FB15k AveSize ↓	FB15k EF ↓
TransE	*KGCP*	*0.096±0.002*	*132.36±6.88*	–	TransE	*KGCP*	*0.131±0.001*	*373.83±2.08*	–
	MCP	0.017±0.001	713.50±180.91	73.56		MCP	0.021±0.000	583.09±10.09	19.02
	CLUSTERCP	0.073±0.007	117.77±8.12	-6.34		CLUSTERCP	0.130±0.000	379.95±2.36	61.20
	APS	0.108±0.001	11428.73±817.03	–		APS	0.154±0.001	1922.92±25.04	–
	RAPS	0.069±0.001	42.03±1.03	–		RAPS	0.124±0.001	336.46±1.10	**-53.39**
	CONDKGCP	0.030±0.001	19.56±0.14	**-17.09**		CONDKGCP	0.027±0.000	78.12±1.23	-28.43
RotatE	*KGCP*	*0.076±0.001*	*2.13±0.25*	–	RotatE	*KGCP*	*0.113±0.001*	*139.57±3.44*	–
	MCP	0.022±0.003	1193.81±382.91	220.68		MCP	0.023±0.000	633.68±9.80	54.90
	CLUSTERCP	0.056±0.001	1.81±0.40	**-0.16**		CLUSTERCP	0.113±0.001	141.56±3.52	–
	APS	0.083±0.002	15963.85±261.07	–		APS	0.128±0.001	1757.13±18.59	–
	RAPS	0.078±0.001	81.96±2.13	–		RAPS	0.122±0.000	416.41±3.46	–
	CONDKGCP	0.045±0.001	2.20±0.61	0.02		CONDKGCP	0.063±0.000	246.80±2.46	**21.45**
RESCAL	*KGCP*	*0.103±0.004*	*2.61±0.15*	–	RESCAL	*KGCP*	*0.092±0.000*	*62.10±0.54*	–
	MCP	0.019±0.003	508.71±118.00	60.25		MCP	0.020±0.001	740.60±30.77	94.24
	CLUSTERCP	0.089±0.009	3.29±0.60	0.49		CLUSTERCP	0.091±0.001	62.25±0.52	**1.50**
	APS	0.106±0.002	1298.86±109.82	–		APS	0.085±0.002	369.19±21.92	483.70
	RAPS	0.074±0.001	43.76±0.46	14.19		RAPS	0.122±0.000	393.44±1.12	–
	CONDKGCP	0.061±0.001	3.80±0.53	**0.28**		CONDKGCP	0.025±0.000	107.91±0.56	6.84
DistMult	*KGCP*	*0.066±0.001*	*2.30±0.05*	–	DistMult	*KGCP*	*0.103±0.001*	*25.16±0.12*	–
	MCP	0.022±0.002	655.33±135.46	148.42		MCP	0.023±0.001	668.83±10.05	80.46
	CLUSTERCP	0.066±0.001	2.32±0.05	–		CLUSTERCP	0.103±0.001	25.60±0.16	–
	APS	0.043±0.002	204.67±39.44	87.99		APS	0.063±0.000	84.82±1.37	14.92
	RAPS	0.065±0.003	51.59±1.04	492.90		RAPS	0.124±0.000	365.76±0.75	–
	CONDKGCP	0.037±0.001	6.18±0.08	**1.34**		CONDKGCP	0.024±0.000	66.27±0.96	**5.20**
ComplEx	*KGCP*	*0.072±0.001*	*1.07±0.01*	–	ComplEx	*KGCP*	*0.088±0.001*	*34.99±0.88*	–
	MCP	0.023±0.002	1898.69±226.32	387.27		MCP	0.024±0.002	664.43±19.46	98.35
	CLUSTERCP	0.072±0.001	1.07±0.01	–		CLUSTERCP	0.088±0.001	34.88±0.85	–
	APS	0.065±0.004	15669.60±498.74	22383.61		APS	0.054±0.001	177.94±10.72	42.04
	RAPS	0.074±0.004	63.82±2.43	–		RAPS	0.121±0.000	417.50±3.41	–
	CONDKGCP	0.049±0.002	1.39±0.01	**0.14**		CONDKGCP	0.026±0.000	166.10±5.23	**21.15**
ConvE	*KGCP*	*0.066±0.004*	*1.71±0.04*	–	ConvE	*KGCP*	*0.102±0.001*	*91.02±7.06*	–
	MCP	0.019±0.002	576.97±170.39	122.40		MCP	0.023±0.000	725.08±38.15	80.26
	CLUSTERCP	0.066±0.001	1.72±0.04	–		CLUSTERCP	0.102±0.001	88.33±7.06	–
	APS	0.071±0.004	9.85±1.77	–		APS	0.110±0.003	4578.40±243.77	–
	RAPS	0.068±0.002	47.75±1.44	–		RAPS	0.123±0.001	400.08±4.54	–
	CONDKGCP	0.038±0.001	4.80±0.10	**1.10**		CONDKGCP	0.032±0.000	429.14±3.11	**48.30**

Table 1: Overall performance comparison of CONDKGCP and baseline methods across six KGE models and two benchmark datasets (WN18 and FB15k). We report CovGap and AveSize as the mean ± standard deviation over 10 independent trials. The EF (efficient rate) is reported as a mean value; its standard deviation is omitted as it is negligible. The best and second-best EF values for each model-dataset pair are highlighted in bold and underline, respectively. KGCP is shown in italic to indicate that it serves as a baseline without conditional coverage guarantees. Our proposed method, CONDKGCP, is highlighted with a gray background. "–" in the EF column denotes a failure case where either CovGap is not reduced or AveSize does not change relative to KGCP.

report the mean and standard deviation.

Evaluation Metrics. Following Ding et al. (2024), we evaluate the performance using two metrics: coverage gap (CovGap) and average size of the prediction sets (AveSize).

Given a set of test triples $\mathcal{T}_{\text{test}}$, the empirical predicate-conditional coverage for each predicate $r \in R$, denoted as Cov_r, is calculated as:

$$\text{Cov}_r = \frac{1}{|\mathcal{T}_{\text{test}}[\{r\}]|} \sum_{(q,e)\in\mathcal{T}_{\text{test}}[\{r\}]} \mathbb{1}[e \in \hat{C}(q)]. \tag{14}$$

The average predicate-conditional coverage gap (CovGap) measures how far the empirical coverage is from the desired coverage level $1-\epsilon$:

$$\text{CovGap} = \frac{1}{|R|} \sum_{r\in R} |\text{Cov}_r - (1-\epsilon)| \tag{15}$$

The average size of the prediction sets (AveSize) is computed as

$$\frac{1}{\mathcal{T}_{\text{test}}[\{r\}]} \sum_{(q,e)\in\mathcal{T}_{\text{test}}[\{r\}]} |\hat{C}(q)|. \tag{16}$$

Note that CovGap and AveSize are inherently competing metrics in conformal prediction (Angelopoulos et al., 2021); reducing CovGap often increases AveSize. For a given CovGap, smaller AveSize is preferred for more informative estimates. Rather than optimizing either metric in isolation,

our aim is to balance the trade-off between coverage probability and prediction set size. To quantify this trade-off, we introduce an auxiliary metric *Efficiency Rate* (ER): the number of additional entities required (relative to KGCP) to reduce CovGap by 0.01:

$$\frac{\text{AvgSize} - \text{AvgSize}^*}{\text{CovGap}^* - \text{CovGap}} \times 0.01, \qquad (17)$$

where $(\text{AvgSize}^*, \text{CovGap}^*)$ are the corresponding values for KGCP.

6.3 Results and Discussion

6.3.1 Overall Comparison

Table 1 presents a comprehensive comparison of CondKGCP with baseline methods across six KGE models and two benchmark datasets (WN18 and FB15k). **Overall, CondKGCP consistently demonstrates the most favorable trade-off between coverage and prediction set size, as evidenced by its lowest EF scores in the majority of cases.**

Among the methods, MCP achieves the lowest CovGap, aligning with Proposition 4.6 in Vovk et al. (2005). Although CovGap is not exactly zero due to the small number of triples for certain subgroups, this deviation is minimal. Thus, MCP's CovGap can be viewed as the empirical lower bound. However, this improved coverage precision comes at a significant cost: MCP produces substantially larger prediction sets, resulting in high AveSize. In contrast, KGCP is designed to achieve only marginal coverage guarantees, resulting in higher CovGap. However, this enables KGCP to generate the smallest prediction sets in most cases, yielding the lowest AveSize empirically. We observe that **CondKGCP achieves CovGap values that are consistently closest to the empirical lower bound, while maintaining compact prediction sets**–often with AveSize values closest to the empirical lower bound-compared to other baseline methods across all evaluated KGE methods and datasets.

Note that while ClusterCP occasionally achieves lower AveSize values comparable to the empirical lower bound, it fails to reduce CovGap in these cases. In fact, both its CovGap and AveSize values remain very close to those of KGCP, the baseline method with marginal coverage guarantees (i.e., calibrated on triples across all predicates), indicating that ClusterCP fails to cluster predicates into meaningful groups. This limitation arises because ClusterCP is designed for subgroups with similar data points, making it unsuitable for our setting, where the distribution of triples across predicates is highly imbalanced.

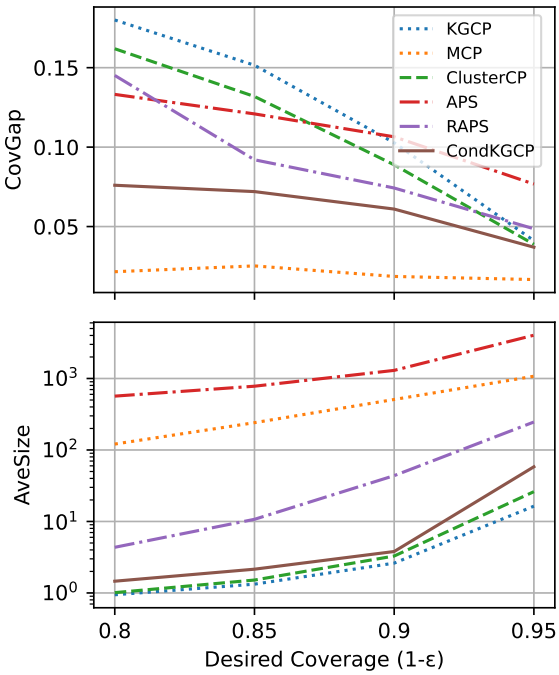

Figure 1: Comparison of methods across varying target coverage levels, showing CovGap (top plot) and AveSize (bottom plot) for RESCAL on WN18. Complete results are provided in Tables 3 and 4 in the Appendix.

Methods that modify the nonconformity score, such as APS and RAPS, often generate overly conservative prediction sets due to prioritization on difficult regions for coverage. As shown in Table 1, neither APS nor RAPS significantly improve CovGap and frequently generate large prediction sets, suggesting the nonconformity measures might not be well-suited for KGE methods.

6.3.2 Comparison across Different Coverage Levels

To evaluate performance under varying coverage levels, we conduct experiments for different target coverage probability $1 - \epsilon \in [0.8, 0.85, 0.9, 0.95]$. The results, shown in Figure 1, reveal that CondKGCP achieves CovGap values closest to the empirical lower bound (MCP) while maintaining AveSize comparable to the empirical lower bound (KGCP) across all coverage levels.

6.3.3 Impact of Hyperparameters

CondKGCP includes two key hyperparameters: ϕ, which controls the granularity of predicate sub-

Model	Method	WN18 CovGap ↓	WN18 AveSize ↓	WN18 EF ↓	Model	Method	FB15k CovGap ↓	FB15k AveSize ↓	FB15k EF ↓
TransE	w/o Merge	0.097±0.002	139.54±6.50	–	TransE	w/o Merge	0.131±0.002	371.89±4.11	–
	w/o RankCal	0.019±0.001	629.13±129.55	791.79		w/o RankCal	0.022±0.000	98.24±2.14	-25.28
	CondKGCP	0.030±0.001	19.56±0.14	**-17.09**		CondKGCP	0.027±0.000	78.12±1.23	**-28.43**
RotatE	w/o Merge	0.077±0.002	1.98±0.54	–	RotatE	w/o Merge	0.114±0.001	141.79±1.32	–
	w/o RankCal	0.023±0.001	1781.28±230.30	335.69		w/o RankCal	0.043±0.000	370.29±3.55	32.96
	CondKGCP	0.045±0.001	2.20±0.61	**0.02**		CondKGCP	0.063±0.000	246.80±2.46	**21.45**
RESCAL	w/o Merge	0.105±0.003	2.57±0.40	–	RESCAL	w/o Merge	0.090±0.002	62.55±0.25	**2.25**
	w/o RankCal	0.021±0.000	385.33±63.64	46.67		w/o RankCal	0.020±0.000	134.22±0.88	10.02
	CondKGCP	0.061±0.001	3.80±0.53	**0.28**		CondKGCP	0.025±0.000	107.91±0.56	6.84
DistMult	w/o Merge	0.066±0.001	3.39±0.04	–	DistMult	w/o Merge	0.103±0.002	25.82±0.21	–
	w/o RankCal	0.023±0.000	719.51±111.22	166.79		w/o RankCal	0.023±0.000	98.00±1.22	9.11
	CondKGCP	0.037±0.001	6.18±0.08	**1.34**		CondKGCP	0.024±0.000	66.27±0.96	**5.20**
ComplEx	w/o Merge	0.074±0.002	1.07±0.01	–	ComplEx	w/o Merge	0.086±0.001	56.49±0.44	107.50
	w/o RankCal	0.025±0.000	2313.94±260.99	492.10		w/o RankCal	0.026±0.000	168.50±5.10	21.53
	CondKGCP	0.049±0.002	1.39±0.01	**0.14**		CondKGCP	0.026±0.000	166.10±5.23	**21.15**
ConvE	w/o Merge	0.066±0.001	2.74±0.05	–	ConvE	w/o Merge	0.099±0.002	180.79±2.79	299.23
	w/o RankCal	0.021±0.000	575.14±89.71	127.43		w/o RankCal	0.027±0.000	579.14±5.03	65.08
	CondKGCP	0.038±0.001	4.80±0.10	**1.10**		CondKGCP	0.032±0.000	429.14±3.11	**48.30**

Table 2: Ablation Study of CondKGCP. Each model is evaluated under three configurations: without predicate merging procedure (w/o Merge), without rank calibration (w/o RankCal), and the proposed CondKGCP (full method).

grouping in the merging procedure, and γ, which balances the conditional coverage guarantee and the size of prediction sets in the dual calibration schema.

As shown in Figure 2, larger values of ϕ result in increased CovGap and reduced AveSize. This is expected, as coarser subgrouping theoretically results in behavior more similar to KGCP, whereas finer subgrouping aligns more closely with MCP.

For γ, larger values are associated with increased AveSize, consistent with the analysis in section 5. However, the empirical results reveal a key practical insight: adjusting γ to sacrifice a small amount in the lower bound of conditional coverage can significantly reduce AveSize while causing only a negligible change in CovGap. This demonstrates the necessity of including γ in the design of CondKGCP.

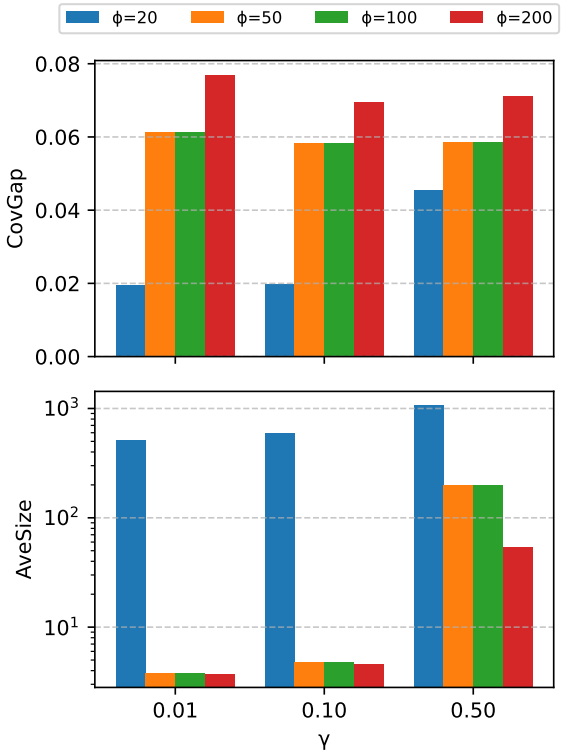

Figure 2: Influence of hyperparameters ϕ and γ on CovGap (top) and AveSize (bottom) for RESCAL on WN18. Complete results for all model-dataset combinations are provided in Tables 5 and 6 in the Appendix.

6.3.4 Impact of Nonconformity Measure

Subgroup-based methods (MCP, ClusterCP, CondKGCP) can be combined with nonconformity score-based methods (APS, RAPS). However, as shown in Table 9 in the Appendix, these combinations do not result in a better trade-off. Notably, CondKGCP achieves low CovGap values with significantly smaller prediction sets in most cases compared to MCP and ClusterCP, regardless of the choice of the nonconformity measure. This observation demonstrates that the improvements offered by CondKGCP are robust to the change of the nonconformity measure.

6.3.5 Impact of Key Components

We compare CONDKGCP with two variants: CONDKGCP without predicate merging procedure (w/o Merge) and CONDKGCP without rank calibration (w/o RankCal). The results in Table 2 show that CONDKGCP outperforms both variants in balancing CovGap and AveSize, achieving the lowest EF and thus the most efficient trade-off. Concretely, CONDKGCP achieves lower CovGap compared to w/o Merge with a comparable AveSize. While it shows slightly higher CovGap than w/o RankCal, it maintains smaller AveSize. This demonstrates the contribution of each component: the merging process effectively reduces CovGap, while rank calibration ensures more compact prediction set.

7 Discussion and Conclusion

In this paper, we introduces CONDKGCP, a novel method that addresses the limitations of existing conformalized KGE method by approximating predicate-conditional coverage guarantees while maintaining compact prediction sets. We theoretically prove that the deviation from the desired confidence level is bounded and empirically demonstrate the effectiveness of CONDKGCP across six KGE methods and two benchmark datasets.

Our method offers a useful uncertainty quantification tool for high-stakes applications, such as medical diagnosis, and can be easily adapted to quantify uncertainty under other types of conditions, such as entity-type. Additionally, it can be seamlessly extended to other tasks, including embedding-based query answering (Ren et al., 2020) and probabilistic reasoning over KG (Zhu et al., 2023, 2024b).

8 Limitation

A potential limitation of the proposed CONDKGCP lies in the probabilistic guarantees provided by Proposition 1, which rely on the assumption of i.i.d. (or weaker exchangeability (Vovk et al., 2005)) data, as well as the assumption that the similarity of vector representations corresponds to the similarity of the distribution of nonconformity scores. While the i.i.d. assumption may occasionally be violated in certain real-world applications, it is a common simplification in statistical methods. As a step forward, we are working on extending our approach to handle covariate shift, where only the input distribution changes while the conditional distribution remains unchanged.

Regarding the similarity assumption, the effectiveness of the predicate merging step, as demonstrated in the ablation study (Table 2), indicates that this assumption is reasonable for KGE methods. Nonetheless, future work could explore incorporating additional features, such as the semantic meaning of predicates, to further enhance the merging process and improve robustness in diverse scenarios.

9 Acknowledgments

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Yuqicheng Zhu. The work was partially supported by EU Projects Graph Massivizer (GA 101093202), enRichMyData (GA 101070284) and SMARTY (GA 101140087), as well as the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1574 – 471687386. Zifeng Ding receives funding from the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation programme grant AVeriTeC (Grant agreement No. 865958).

References

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

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2).

Russa Biswas, Lucie-Aimée Kaffee, Michael Cochez, Stefania Dumbrava, Theis E Jendal, Matteo Lissandrini, Vanessa Lopez, Eneldo Loza Mencía, Heiko Paulheim, Harald Sack, et al. 2023. Knowledge graph embeddings: open challenges and opportunities. *Transactions on Graph Data and Knowledge*, 1(1):4–1.

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.

Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. 2020. LibKGE - A knowledge graph embedding library for reproducible research. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 165–174.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Tiffany Ding, Anastasios Angelopoulos, Stephen Bates, Michael Jordan, and Ryan J Tibshirani. 2024. Class-conditional conformal prediction with many classes. *Advances in Neural Information Processing Systems*, 36.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. 2021. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482.

Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. 2015. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 623–632.

Yunjie He, Daniel Hernandez, Mojtaba Nayyeri, Bo Xiong, Yuqicheng Zhu, Evgeny Kharlamov, and Steffen Staab. 2024. Generating $sroi^-$ ontologies via knowledge graph query embedding learning. In *Proceeding of 27th European Conference on Artificial Intelligence*, pages 4279 – 4286.

Yunjie He, Bo Xiong, Daniel Hernández, Yuqicheng Zhu, Evgeny Kharlamov, and Steffen Staab. 2025. Dage: Dag query answering via relational combinator with logical constraints. In *Proceedings of the ACM on Web Conference 2025*, pages 2514–2529.

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.

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.

H Ren, W Hu, and J Leskovec. 2020. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In *International Conference on Learning Representations (ICLR)*.

Yaniv Romano, Matteo Sesia, and Emmanuel Candes. 2020. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2019. You can teach an old dog new tricks! on training knowledge graph embeddings. In *International Conference on Learning Representations*.

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.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4498–4507.

Yuanjie Shi, Subhankar Ghosh, Taha Belkhouja, Janardhan Rao Doppa, and Yan Yan. 2024. Conformal prediction for class-wise coverage via augmented label rank calibration. In *NeurIPS*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *ICLR (Poster)*. OpenReview.net.

Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent knowledge graph embedding for effective recommendation. In *Proceedings of the 12th ACM conference on recommender systems*, pages 297–305.

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

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pages 57–66.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR.

Vladimir Vovk. 2012. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pages 475–490. PMLR.

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

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE transactions on knowledge and data engineering*, 29(12):2724–2743.

Bo Xiong, Shichao Zhu, Mojtaba Nayyeri, Chengjin Xu, Shirui Pan, Chuan Zhou, and Steffen Staab. 2022. Ultrahyperbolic knowledge graph embeddings. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2130–2139.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2018. One-shot relational learning for knowledge graphs. In *EMNLP*, pages 1980–1990. Association for Computational Linguistics.

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

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. *arXiv preprint arXiv:2410.15981*.

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. 2025. 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. 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. *arXiv preprint arXiv: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 Proofs

Proposition 1 (Conditional Coverage Guarantee)**.** *Let q be a query and e be its answer entity. Given a user-specified error rate ϵ and a $\gamma \in [0, 1]$, we have the following bounds for all parts $g \in P$:*

$$\mathbb{P}(e \in \hat{C}_{\text{CondKGCP}}(q) \mid \text{pred}(q) \in g) \geq 1 - \epsilon - (1 - \gamma)\epsilon_g^{\hat{k}(g)}, \tag{18}$$

and if there is no tie in the set of nonconformity scores of the triples in $\mathcal{T}_g$, then

$$\mathbb{P}(e \in \hat{C}_{\text{CondKGCP}}(q) \mid \text{pred}(q) \in g) \leq 1 - \epsilon + \gamma\epsilon_g^{\hat{k}(g)} + \frac{1}{|\mathcal{T}_g| + 1}. \tag{19}$$

Proof of the lower bound. We prove the lower bound similar to the proof of Theorem 4.1 in Shi et al. (2024, Appendix A.1). We first conduct conformal prediction for each part $g \in P$ (which can be viewed as applying MCP where each subgroup contains triples whose predicates are in a predicate set generated by Algorithm 1) with the adjust error rate $\epsilon'(g)$. The prediction set is defined as follows:

$$\hat{C}_{\text{MCP}^*}(q) = E_q[S \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g)] \tag{20}$$

According to (Vovk et al., 2005, Proposition 4.6), for a query q, we have

$$\mathbb{P}(e \in \hat{C}_{\text{MCP}^*}(q) \mid \text{pred}(q) \in g) \geq 1 - \epsilon' \tag{21}$$

Suppose that the rank threshold for part g is $\hat{k}(g)$ and $\epsilon_g^{\hat{k}(g)}$ is its corresponding miscoverage error of top-$\hat{k}(g)$ prediction sets, we have

$$\begin{aligned}
&\mathbb{P}(e \in \hat{C}_{\text{MCP}^*}(q) \mid \text{pred}(q) \in g) \\
=&\mathbb{P}\Big(S(tr(q,e)) \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g) \mid \text{pred}(q) \in g\Big) \\
=&\mathbb{P}\Big(S(tr(q,e)) \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g), \text{rank}_{M_\theta}(q,e) \leq \hat{k}(g) \mid \text{pred}(q) \in g\Big) \\
&+ \mathbb{P}\Big(S(tr(q,e)) \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g), \text{rank}_{M_\theta}(q,e) > \hat{k}(g) \mid \text{pred}(q) \in g\Big) \\
\leq&\mathbb{P}\Big(S(tr(q,e)) \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g), \text{rank}_{M_\theta}(q,e) \leq \hat{k}(g) \mid \text{pred}(q) \in g\Big) \\
&+ \underbrace{\mathbb{P}(\text{rank}_{M_\theta}(q,e) > \hat{k}(g) \mid \text{pred}(q) \in g)}_{\hat{k}(g)}
\end{aligned}$$

By definition of the prediction set constructed by CONDKGCP, we have

$$\mathbb{P}(e \in \hat{C}_{\text{CondKGCP}}(q) \mid \text{pred}(q) \in g) \geq 1 - \epsilon'(g) - \epsilon_g^{\hat{k}(g)} \tag{22}$$

In our paper we set $\epsilon'(g) = \epsilon - \gamma \cdot \epsilon_g^{\hat{k}(g)}$, therefore, we have

$$\mathbb{P}(e \in \hat{C}_{\text{CondKGCP}}(q) \mid \text{pred}(q) \in g) \geq 1 - \epsilon - (1 - \gamma)\epsilon_g^{\hat{k}(g)} \tag{23}$$

□

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 of the triples in $\mathcal{T}_g$, denoted as $\mathcal{S}(\mathcal{T}_g)$, we know that the nonconformity scores in $\mathcal{S}(\mathcal{T}_g)$ are all distinct with probability one. The set $\hat{C}_{\text{MCP}^*}(q)$ is equivalent to the set of all answer entity $e' \in E$ such that the nonconformity score $S(tr(q, e'))$ ranks among the $\lceil(|\mathcal{T}_g| + 1)(1 - \epsilon'(g))\rceil$ smallest of $\mathcal{S}(\mathcal{T}_g)$.

Consider now the complementary set $\hat{D}_{\text{MCP}^*}(q)$ consisting of answer entities $e' \in E$ such that the nonconformity score $S(tr(q, e'))$ is among the $\lceil (|\mathcal{T}_g| + 1)\epsilon'(g) - 1 \rceil$ largest. Under i.i.d assumption (or a weaker exchangeability assumption), the joint distribution of the nonconformity scores $\mathcal{S}(\mathcal{T}_g)$ is invariant under permutations. As a result, the ranks of the nonconformity scores in $\mathcal{S}(\mathcal{T}_g)$ are uniformly distributed among $\{1, 2, \ldots, |\mathcal{T}_g|\}$ and hence we can derive the following lower bound for each part $g \in P$:

$$\mathbb{P}(e \in \hat{D}_{\text{MCP}^*}(q) \mid \text{pred}(q) \in g) \geq \epsilon'(g) - \frac{1}{|\mathcal{T}_g| + 1}, \tag{24}$$

We also know that there is no intersected elements for $\hat{C}_{\text{MCP}^*}(q)$ and $\hat{D}_{\text{MCP}^*}(q)$

$$\hat{C}_{\text{MCP}^*}(q) \cap \hat{D}_{\text{MCP}^*}(q) = \emptyset \tag{25}$$

Then we can derive the upper bound for each part g as follows:

$$\begin{aligned}
&\mathbb{P}(e \in \hat{C}_{\text{MCP}^*}(q) \mid \text{pred}(q) \in g) + \mathbb{P}(e \in \hat{D}_{\text{MCP}^*}(q) \mid \text{pred}(q) \in g) \leq 1 \\
\Rightarrow &\mathbb{P}(e \in \hat{C}_{\text{MCP}^*}(q) \mid \text{pred}(q) \in g) \leq 1 - \mathbb{P}(e \in \hat{D}_{\text{MCP}^*}(q) \mid \text{pred}(q) \in g) \\
\Rightarrow &\mathbb{P}(e \in \hat{C}_{\text{MCP}^*}(q) \mid \text{pred}(q) \in g) \leq 1 - \epsilon'(g) + \frac{1}{|\mathcal{T}_g| + 1} \\
\Rightarrow &\mathbb{P}(e \in \hat{C}_{\text{MCP}^*}(q) \mid \text{pred}(q) \in g) \leq 1 - \epsilon + \gamma \epsilon_g^{\hat{k}(g)} + \frac{1}{|\mathcal{I}^g| + 1}
\end{aligned}$$

Based on the definition of CONDKGCP, we have

$$\begin{aligned}
&\mathbb{P}(e \in \hat{C}_{\text{MCP}^*}(q) \mid \text{pred}(q) \in g) \\
=&\underbrace{\mathbb{P}\Big(S(tr(q, e)) \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g), \text{rank}_{M_\theta}(q, e) \leq \hat{k}(g) \mid \text{pred}(q) \in g\Big)}_{\mathbb{P}(e \in \hat{C}_{\text{CondKGCP}}(q) | \text{pred}(q) \in g)} \\
&+ \mathbb{P}\Big(S(tr(q, e)) \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g), \text{rank}_{M_\theta}(q, e) > \hat{k}(g) \mid \text{pred}(q) \in g\Big) \\
\Rightarrow &\mathbb{P}(e \in \hat{C}_{\text{CondKGCP}}(q) \mid \text{pred}(q) \in g) \leq \mathbb{P}(e \in \hat{C}_{\text{MCP}^*}(q) \mid \text{pred}(q) \in g) \\
\Rightarrow &\mathbb{P}(e \in \hat{C}_{\text{CondKGCP}}(q) \mid \text{pred}(q) \in g) \leq 1 - \epsilon + \gamma \epsilon_g^{\hat{k}(g)} + \frac{1}{|\mathcal{T}_g| + 1}
\end{aligned}$$

□

Corollary 2 (Shi et al. (2024)). *Suppose $\epsilon'(g)$ and $\hat{k}(g)$ satisfy*

$$\hat{k}(g) \in \{k : \epsilon_g^{\hat{k}(g)} < \epsilon\}; 0 \leq \epsilon'(g) \leq \epsilon - \epsilon_g^{\hat{k}(g)}, \tag{26}$$

the rank calibration guarantee to shrink the prediction sets, if for a query q and any $e' \in E$:

$$\mathbb{P}_q\Big(S(tr(q, e')) \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g), \text{rank}_{M_\theta}(q, e') \leq \hat{k}(g)\Big) \leq \mathbb{P}_q\Big(S(tr(q, e')) \leq \hat{s}_\epsilon(\mathcal{T}_g)\Big) \tag{27}$$

Proof. We can prove this Corollary based on Shi et al. (2024, Appendix A.2).Define the following fraction:

$$\sigma_g = \frac{\mathbb{P}_q\Big(S(tr(q, e')) \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g), \text{rank}_{M_\theta}(q, e') \leq \hat{k}(g)\Big)}{\mathbb{P}_q(S(tr(q, e')) \leq \hat{s}_\epsilon(\mathcal{T}_g)}. \tag{28}$$

By the assumption in Equation (27), it follows that $\sigma_g \leq 1$. The expected size of the prediction set for CONDKGCP is given by:

$$\mathbb{E}_q\Big[|\hat{C}_{\text{CondKGCP}}(q)|\Big] = \mathbb{E}_q\Bigg[\sum_{e' \in E} \mathbb{1}\Big[S(tr(q,e')) \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g), \text{rank}_{M_\theta}(q,e') \leq \hat{k}(g)\Big]\Bigg] \tag{29}$$

$$= \sum_{e' \in E} \mathbb{E}_q\Big[\mathbb{1}\Big[S(tr(q,e')) \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g), \text{rank}_{M_\theta}(q,e') \leq \hat{k}(g)\Big]\Big] \tag{30}$$

$$= \sum_{e' \in E} \mathbb{P}_q\Big(S(tr(q,e')) \leq \hat{s}_{\epsilon'(g)}(\mathcal{T}_g), \text{rank}_{M_\theta}(q,e') \leq \hat{k}(g)\Big) \tag{31}$$

By the definition of σ_g and the assumption $\sigma_g \leq 1$, we obtain:

$$\mathbb{E}_q\Big[|\hat{C}_{\text{CondKGCP}}(q)|\Big] = \sum_{e' \in E} \sigma_g \cdot \mathbb{P}_q\Big(S(tr(q,e')) \leq \hat{s}_\epsilon(\mathcal{T}_g)\Big) \tag{32}$$

$$\leq \sum_{e' \in E} \mathbb{E}_q\Big[\mathbb{1}\Big[S(tr(q,e')) \leq \hat{s}_\epsilon(\mathcal{T}_g)\Big]\Big] \tag{33}$$

$$= \mathbb{E}_q\Bigg[\sum_{e' \in E} \mathbb{1}\Big[S(tr(q,e')) \leq \hat{s}_\epsilon(\mathcal{T}_g)\Big]\Bigg] \tag{34}$$

$$= \mathbb{E}_q\Big[|\hat{C}_{\text{MCP}^*}(q)|\Big] \tag{35}$$

Therefore, we conclude that adding rank calibration always reduces the prediction set size, provided that the condition in Equation (27) holds. □

B Discussion About Merging Process

Note the underlying idea of predicate merging procedure in section 4.1 is essentially very similar to CLUSTERCP proposed by recent work (Ding et al., 2024), where subgroups are clustered based on similarity of the nonconformity score distribution. But in our case, we have extremely imbalanced size of calibration data for each predicate. For many predicates, the number of calibration triples is too small to capture the characteristics of score distribution with quantile vectors proposed in Ding et al. (2024), thus resulting unreliable clustering and limited improvement in terms of conditional coverage probability. We can see in Table 1 that CLUSTERCP fails to provide meaningful fine-grained predicate clustering, reflected by very similar performance as KGCP (baseline method that is not designed to achieve conditional coverage guarantees).

C Verification of the Condition in Equation (13)

As we prove in Appendix A, CONDKGCP always has smaller expected prediction set size compared to CONDKGCP without rank calibration, i.e., MCP at set part level under the condition in Equation (13). In this section, we empirically verify the condition in Equation (13) across all model-dataset combinations.

We use two metrics to verify the condition. Condition Satisfaction Rate (CSR) quantifies how often the condition $\sigma_g \leq 1$ holds for all $g \in P$:

$$CSR := \sum_{g \in P} \mathbb{1}[\sigma_g \leq 1]; \tag{36}$$

And $\bar{\sigma}$ computes the average value of σ_g for all $g \in P$, denoted as $\bar{\sigma}$:

$$\bar{\sigma} := \frac{1}{|P|} \sum_{g \in P} \sigma_g. \tag{37}$$

The results of these two metrics for all model-dataset combinations are reported in Table 3. The results show that the condition $\sigma_g \leq 1$ holds for nearly all $g \in P$ under different model-dataset combinations. This provides empirical evidence that the additional rank calibration schema reduces the size of prediction sets. Moreover, we observe that smaller values of $\bar{\sigma}$ are typically associated with smaller AveSize in in Table 1.

	WN18		FB15k	
Model	CSR (%)	$\bar{\sigma}$	CSR (%)	$\bar{\sigma}$
TransE	91.7	0.823	98.7	0.633
RotatE	100	0.422	97.6	0.899
RESCAL	100	0.554	99.3	0.712
DistMult	91.7	0.783	99.4	0.512
ComplEx	100	0.401	96.4	0.703
ConvE	100	0.555	93.1	0.951

Table 3: Verification of Condition in Equation (13).

D SOFTMAX Nonconformity Score

By default we use SOFTMAX nonconformity score defined in Zhu et al. (2025):

$$S(tr(q,e)) = 1 - \hat{M}_\theta(tr(q,e)),$$

where

$$\hat{M}_\theta(tr(q,e)) = \frac{\exp(M_\theta(tr(q,e)))}{\sum_{e' \in E} \exp\left(M_\theta(q,e')\right)}.$$

E APS and RAPS Nonconformity Score

Adaptive Predication Sets (APS) (Romano et al., 2020) improves the conditional coverage probability by modifying the nonconformity measure. Specifically, given a query q, for all $e \in E$, we first normalize the plausibility score using softmax function:

$$\hat{M}_\theta(tr(q,e)) = \frac{\exp(M_\theta(tr(q,e)))}{\sum_{e' \in E} \exp\left(M_\theta(q,e')\right)}.$$

Then we sort the normalized scores such that $1 \geq \hat{M}_{(1)} \geq \cdots \geq \hat{M}_{(|E|)}$, where $M_{(k)}$ denotes the k-th largest plausibility score. Recall that $\text{rank}_{M_\theta}(q,e)$ denotes the rank of e given query q. The nonconformity score of APS is then defined as

$$S(tr(q,e)) = \sum_{i=1}^{\text{rank}_{M_\theta}(q,e)-1} \hat{M}_{(i)} + U \cdot \hat{M}_{(\text{rank}_{M_\theta}(q,e))},$$

where $U \in [0,1]$ is a uniform random variable.

The regularized version - RAPS additionally includes a rank-based regularization term to the nonconformity score.

$$S(tr(q,e)) = \sum_{i=1}^{\text{rank}_{M_\theta}(q,e)-1} \hat{M}_{(i)} + U \cdot \hat{M}_{(\text{rank}_{M_\theta}(q,e))} + \lambda \cdot \max\{\text{rank}_{M_\theta}(q,e) - k_{reg}, 0\},$$

where λ and k_{reg} are two hyper-parameters.

	#Entity	#Relation	#Training	#Validation	#Test
WN18	40,943	18	141,442	5,000	5,000
WN18RR	40,943	11	86,835	3,034	3,134
FB15k	14,951	1,345	483,142	50,000	59,071
FB15k-237	14,541	237	272,115	17,535	20,466

Table 4: Statistics of benchmark datasets for link prediction task.

	Scoring Function $s(< h, r, t >)$
TransE (Bordes et al., 2013)	$-\|\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|\|_{1/2}$
RotatE (Sun et al., 2019)	$-\|\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|\|_{p}$
RESCAL (Nickel et al., 2011)	$\mathbf{h}^T \mathbf{M}_r \mathbf{t}$
DistMult (Yang et al., 2015)	$\mathbf{h}^T diag(\mathbf{r}) \mathbf{t}$
ComplEx (Trouillon et al., 2016)	$Re(\mathbf{h}^T diag(\mathbf{r}) \bar{\mathbf{t}})$
ConvE (Dettmers et al., 2018)	$f(vec(f([\bar{\mathbf{h}}; \bar{\mathbf{r}}] * \omega))\mathbf{W})\mathbf{t}$

Table 5: The scoring function $s(< \boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t} >)$ of KGE models used in this paper, where $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}$ denote the embeddings of h, r, t, $\circ$ denotes Hadamard product. $\bar{\cdot}$ refers to conjugate for complex vectors in ComplEx, and 2D reshaping for real vectors in ConvE. $*$ is operator for 2D convolution. ω is the filters and W is the parameters for 2D convolutional layer.

F Detailed Experimental Settings

Note the experimental settings closely follow the approach outlined by Zhu et al. (2025). For completeness and to ensure the paper is self-contained, we recall the details in this section.

F.1 Information About KGE Models and Benchmark Datasets

Table 4 outlines key statistics for the benchmark datasets, while Table 5 presents the scoring functions utilized by various KGE methods.

F.2 Privacy Concerns in FB15k and FB15k-237

Both FB15k and FB15k-237 datasets include data about individuals, predominantly well-known public figures such as politicians, celebrities, and historical icons. Since this information is widely accessible through public platforms and online sources, its inclusion in Freebase poses minimal privacy risks compared to datasets containing sensitive or private personal details.

F.3 Details of Pre-training KGE Models

For pre-training the the KGE models, we follow (Zhu et al., 2025).

The LibKGE framework (Broscheit et al., 2020) was used for training the KGE models, following a hyperparameter optimization approach inspired by (Ruffinelli et al., 2019). The experiments were executed on a Linux system equipped with a 40GB NVIDIA A100 SXM4 GPU.

Initially, we applied a quasi-random hyperparameter search using Sobol sequences to ensure an even distribution of configurations, avoiding clustering (Bergstra and Bengio, 2012). For each combination of dataset, model, training type, and loss function, 30 configurations were generated. This was followed by a Bayesian optimization phase, incorporating 30 additional trials to refine the hyperparameters based on prior results. The Ax framework (`https://ax.dev/`) facilitated this process.

The search spanned a comprehensive hyperparameter space, encompassing loss functions (pairwise margin ranking with hinge loss, binary cross-entropy, cross-entropy), regularization methods (none/L1/L2/L3, dropout), optimizers (Adam, Adagrad), and common initialization techniques used in the KGE domain. Embedding sizes of 128, 256, and 512 were considered. For further details, refer to (Ruffinelli et al., 2019, Table 5).

Configuration files for baseline and competing models, along with models used in aggregation, are available in the "configs" folder of the submitted software directory. These files (*.yaml) document the hyperparameter settings applied in this study.

F.4 Optimal Hyperparameter Settings

The optimal hyperparameter configurations for each model-dataset combination are summarized in Table 7.

G Complete Experimental Results

In Figure 3 and 4, we show the complete results of comparison of methods' CovGap and AveSize across varying target coverage levels ranging from $[0.8, 0.85, 0.9, 0.95]$.

In Figure 5 and 6, we investigate the influence of hyperparameters ϕ and γ on CovGap and AveSize, respectively.

H Complexity Analysis

The total computational cost of our method, CondKGCP, comprises three main components:

- **KGE Model Training**: This is shared across all methods and is the most computationally intensive component.
- **Calibration Step**: This includes any method-specific operations, such as clustering in

Method	WN18 Training	WN18 Calibration	WN18 Set Construction	Method	FB15k Training	FB15k Calibration	FB15k Set Construction
KGCP	1h	6.68s	0.8ms/query	KGCP	2h	32.70s	0.4ms/query
MCP	1h	7.32s	0.8ms/query	MCP	2h	32.70s	0.4ms/query
CLUSTERCP	1h	8.30s	0.8ms/query	CLUSTERCP	2h	33.55s	0.4ms/query
APS	1h	7.28s	1ms/query	APS	2h	32.56s	0.4ms/query
RAPS	1h	8.72s	1ms/query	RAPS	2h	39.23s	0.5ms/query
CONDKGCP	1h	8.92s	1ms/query	CONDKGCP	2h	38.33s	0.5ms/query

Table 6: Empirical runtime for different uncertainty quantification methods on WN18 and FB15k.

Model	WN18 gamma	WN18 cut	FB15k gamma	FB15k cut
TransE	0.01	50	0.01	20
RotatE	0.01	50	0.01	100
RESCAL	0.01	50	0.01	50
DistMult	0.01	50	0.01	20
ComplEx	0.1	50	0.01	20
ConvE	0.01	50	0.01	50

Table 7: Optimal hyperparameter configurations for CONDKGCP across various model-dataset combinations.

CLUSTERCP or dual calibration in CONDKGCP. These are one-time offline steps and incur negligible overhead relative to model training.

- **Test-Time Set Construction**: This is the only component executed per test query and constitutes the primary source of runtime differences between methods.

We now present a theoretical complexity analysis focused on the per-query cost:

- Baseline methods (e.g., KGCP, MCP): Each query requires computing scores for all candidate entities, with time complexity $\mathcal{O}(|E| \cdot d)$, where $|E|$ is the number of entities and d is the embedding dimension. An additional $\mathcal{O}(|E|)$ is needed for score thresholding to form the prediction set.

- CONDKGCP: In addition to score computation ($\mathcal{O}(|E| \cdot d)$), our method includes rank-based thresholding from dual calibration, adding a sorting step $\mathcal{O}(|E| \log |E|)$, linear pass $\mathcal{O}(|E|)$, and selection of the top-K entities ($\mathcal{O}(K)$). Thus, the total per-query complexity is: $\mathcal{O}(|E| \cdot d) + \mathcal{O}(|E| \log |E| + |E| + K)$.

	CovGap (0.95)	AveSize (0.95)	Ratio (0.95)
KGCP	0.040	16.30	–
MCP	0.013	1128.46	41191.11
CLUSTERCP	0.040	19.61	–
APS	0.079	4036.71	–
RAPS	0.051	242.36	–
CONDKGCP	0.029	133.37	10642.73
	CovGap (0.98)	AveSize (0.98)	Ratio (0.98)
KGCP	0.020	2945.21	–
MCP	0.009	5027.66	189313.64
CLUSTERCP	0.022	2933.73	–
APS	0.039	12094.28	–
RAPS	0.018	3985.23	520010.00
CONDKGCP	0.017	3060.69	38493.33
	CovGap (0.99)	AveSize (0.99)	Ratio (0.99)
KGCP	0.016	7403.45	–
MCP	0.006	9494.30	209085.00
CLUSTERCP	0.018	8921.37	–
APS	0.016	19486.37	–
RAPS	0.009	17098.86	1385058.57
CONDKGCP	0.013	7614.11	70220.00

Table 8: Performance results at high confidence levels ($1 - \epsilon > 0.95$).

While CONDKGCP introduces additional steps beyond baselines, the dominant term remains $\mathcal{O}(|E| \cdot d)$ in practice. For instance, with $|E| = 10^6$ and $d = 512$, computing scores dominates (512 operations per entity), whereas the extra overhead from sorting and filtering (approximately 21 operations per entity) is comparatively negligible.

Empirical runtimes, reported in Table 6, confirm that the additional complexity of CONDKGCP does not translate into significant runtime overhead on a NVIDIA A100 GPU.

I Results for More Confidence Levels

In Table 8, we extend the analysis from Figure 1 to higher confidence levels (i.e., $\alpha > 0.95$) using the same experimental settings. We observe that CONDKGCP consistently achieves superior conditional coverage while maintaining a more favorable trade-off between coverage and prediction set size,

even at these stringent confidence levels.

Notably, all methods exhibit a sharp increase in prediction set size beyond $\alpha = 0.95$. We attribute this to the inherent limitations of the base KGE models. For instance, on WN18, most models attain Hits@10 close to 0.95, meaning that approximately 95% of correct answers are ranked within the top 10. Pushing coverage beyond this threshold necessitates including lower-ranked (and often noisy) entities, which substantially enlarges the prediction sets.

This observation underscores an important practical consideration: in high-stakes applications where coverage above 95% is required, it is essential to pair uncertainty quantification with a highly accurate base model (e.g., achieving Hits@$K >$ 0.95). Otherwise, the resulting prediction sets may become impractically large.

J AI Assistants In Writing

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

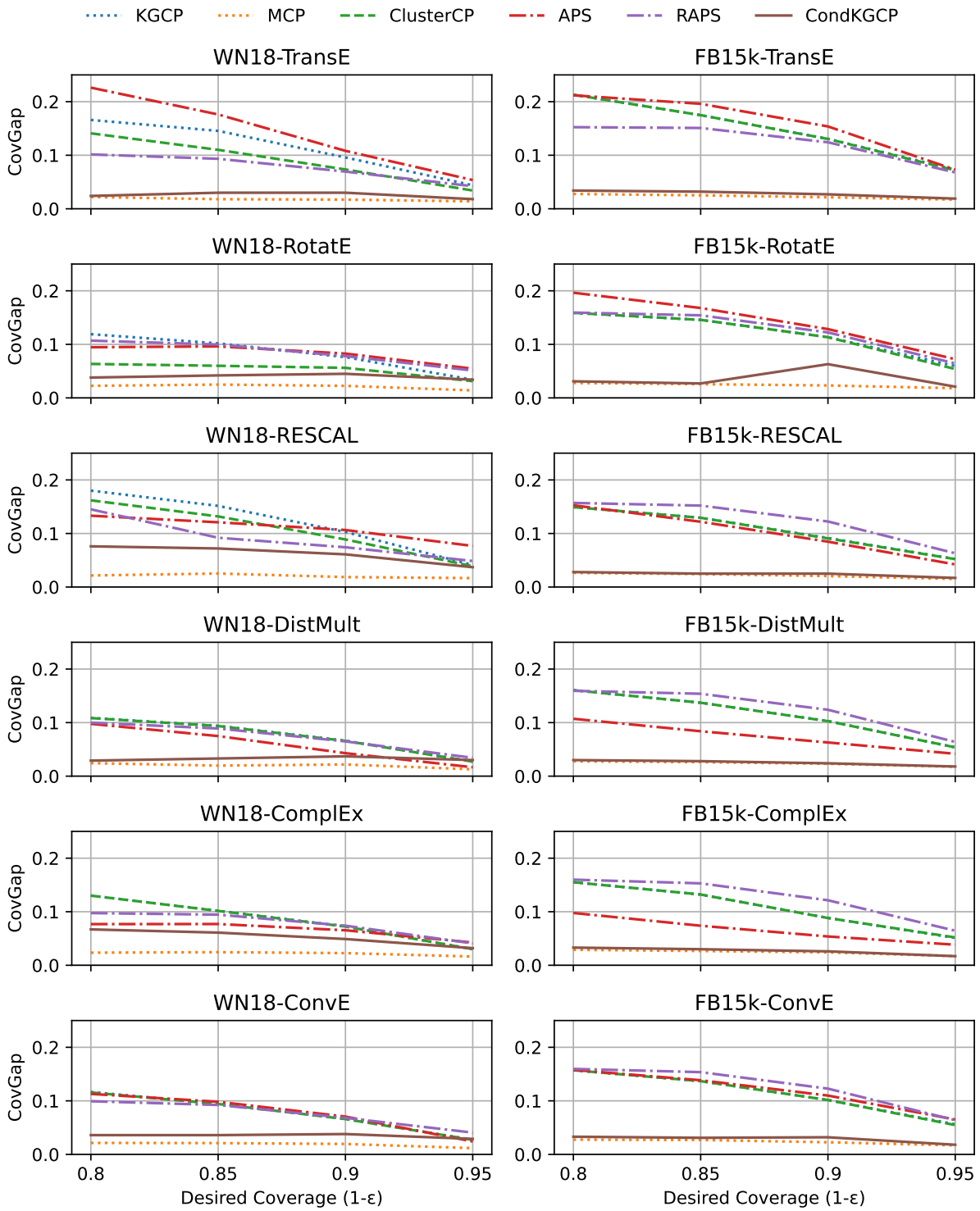

Figure 3: Complete results of comparison of methods' CovGap across varying target coverage levels.

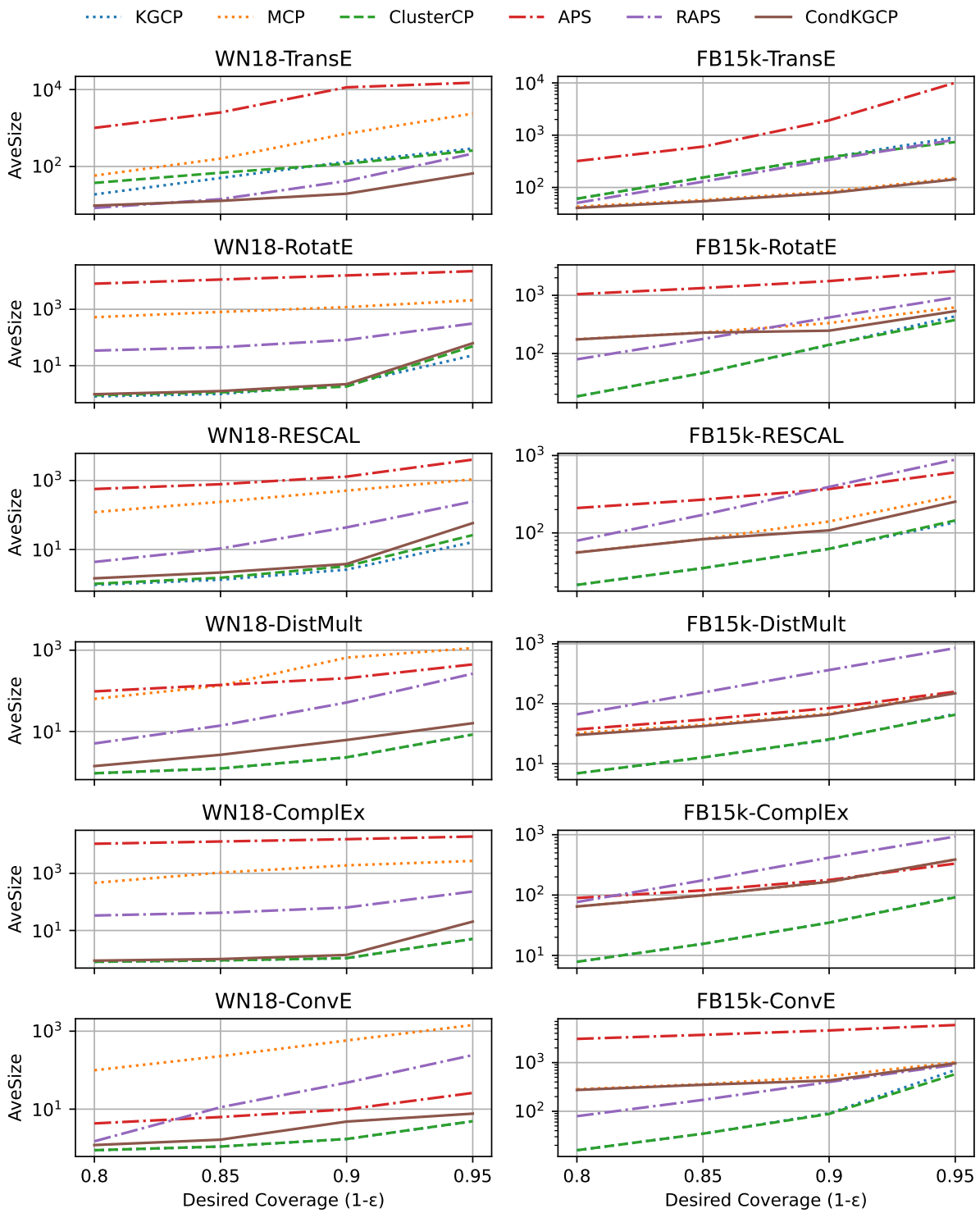

Figure 4: Complete results of comparison of methods' AveSize across varying target coverage levels.

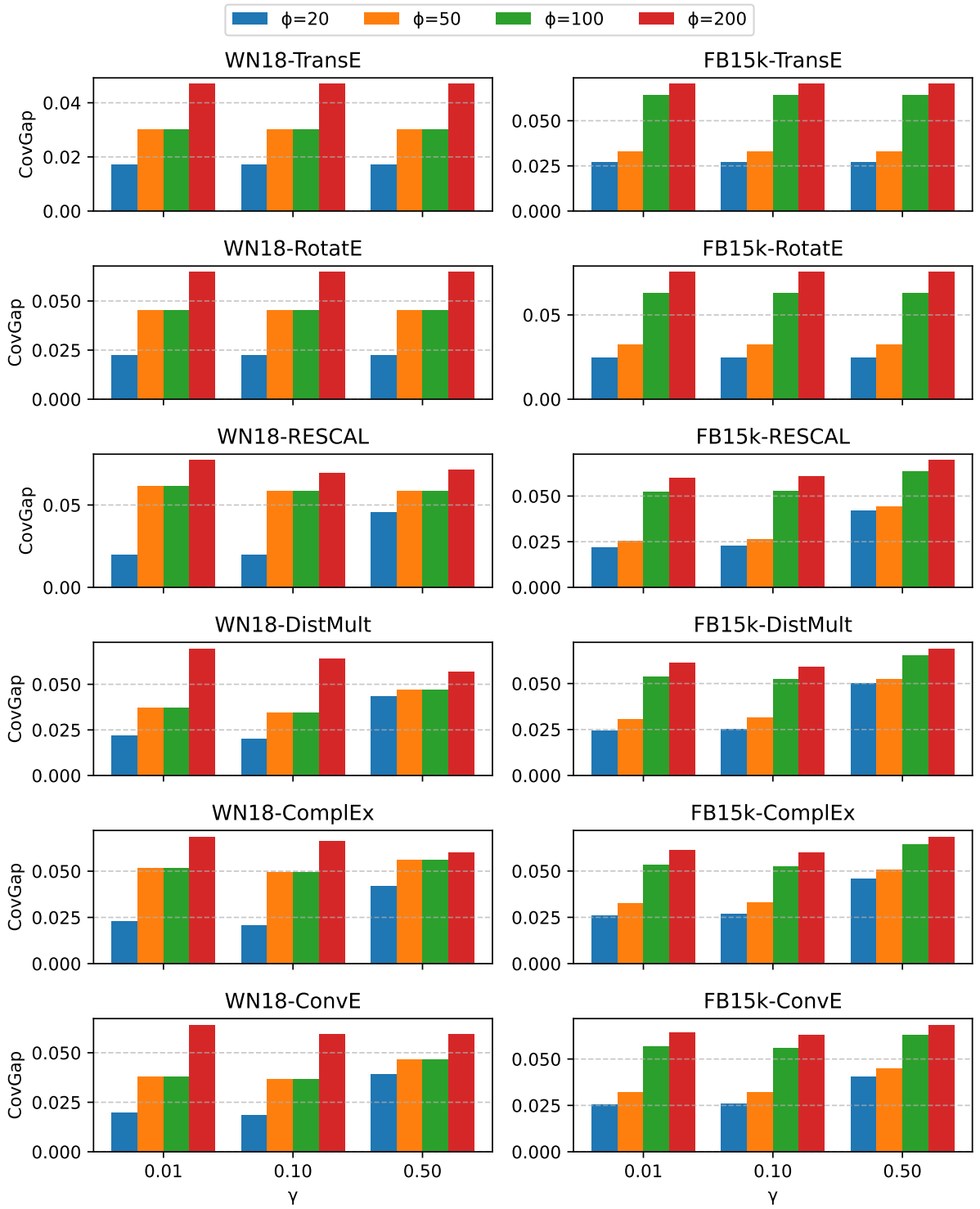

Figure 5: Influence of hyperparameters ϕ and γ on CovGap.

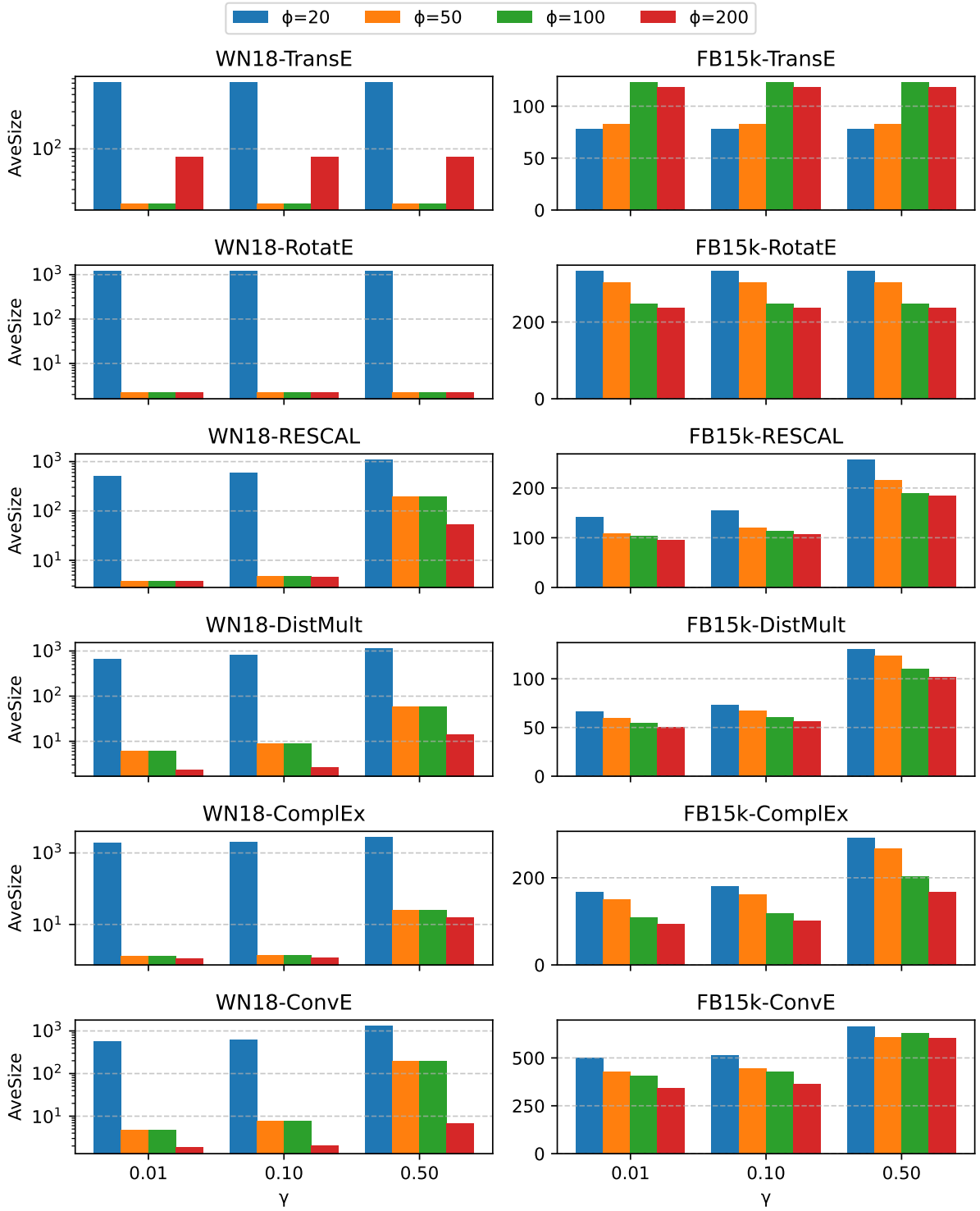

Figure 6: Influence of hyperparameters ϕ and γ on AveSize.

Model	Score	WN18 Method	CovGap ↓	AveSize ↓	Model	Score	FB15k Method	CovGap ↓	AveSize ↓
TransE	APS	MCP	0.015	1320.77	TransE	APS	MCP	0.022	653.62
		ClusterCP	0.079	9081.71			ClusterCP	0.116	4925.66
		CondKGCP	0.016	1090.23			CondKGCP	0.028	158.17
	RAPS	MCP	0.024	1172.74		RAPS	MCP	0.021	566.15
		ClusterCP	0.066	1138.93			ClusterCP	0.122	320.57
		CondKGCP	0.051	50.29			CondKGCP	0.023	258.90
RotatE	APS	MCP	0.014	15922.23	RotatE	APS	MCP	0.022	1695.87
		ClusterCP	0.077	15398.99			ClusterCP	0.122	1626.64
		CondKGCP	0.014	15920.71			CondKGCP	0.023	1389.67
	RAPS	MCP	0.022	2084.58		RAPS	MCP	0.022	523.75
		ClusterCP	0.063	2051.17			ClusterCP	0.121	398.75
		CondKGCP	0.039	141.42			CondKGCP	0.023	322.63
RESCAL	APS	MCP	0.020	3236.86	RESCAL	APS	MCP	0.021	867.08
		ClusterCP	0.095	1932.13			ClusterCP	0.072	360.10
		CondKGCP	0.020	3334.02			CondKGCP	0.024	552.58
	RAPS	MCP	0.020	1037.90		RAPS	MCP	0.021	590.23
		ClusterCP	0.060	1001.70			ClusterCP	0.121	374.67
		CondKGCP	0.035	94.29			CondKGCP	0.025	287.15
DistMult	APS	MCP	0.018	815.22	DistMult	APS	MCP	0.024	723.68
		ClusterCP	0.043	205.65			ClusterCP	0.063	85.04
		CondKGCP	0.019	249.74			CondKGCP	0.024	124.64
	RAPS	MCP	0.022	1003.06		RAPS	MCP	0.022	576.24
		ClusterCP	0.059	969.60			ClusterCP	0.123	356.66
		CondKGCP	0.024	140.75			CondKGCP	0.023	273.34
ComplEx	APS	MCP	0.017	16707.74	ComplEx	APS	MCP	0.024	694.64
		ClusterCP	0.065	15738.09			ClusterCP	0.054	176.78
		CondKGCP	0.017	16728.70			CondKGCP	0.025	282.54
	RAPS	MCP	0.023	2614.74		RAPS	MCP	0.023	530.80
		ClusterCP	0.064	2598.85			ClusterCP	0.120	401.65
		CondKGCP	0.041	83.65			CondKGCP	0.026	331.38
ConvE	APS	MCP	0.015	589.50	ConvE	APS	MCP	0.022	3951.64
		ClusterCP	0.070	9.91			ClusterCP	0.109	4376.19
		CondKGCP	0.052	11.89			CondKGCP	0.025	3019.98
	RAPS	MCP	0.023	1013.09		RAPS	MCP	0.022	613.67
		ClusterCP	0.060	981.74			ClusterCP	0.121	386.85
		CondKGCP	0.033	116.70			CondKGCP	0.025	304.28

Table 9: Overall performance comparison of MCP, ClusterCP, and CondKGCP using different nonconformity measures: APS and RAPS. Note that methods modifying nonconformity scores theoretically can be combined with subgroup-based methods, however since APS and RAPS are not suitable for KGE methods we do not see much improvement by combining these methods, nevertheless, CondKGCP outperforms both MCP and ClusterCP if we combine with APS or RAPS across all evaluation metrics.