Predictive Multiplicity of Knowledge Graph Embeddings in Link Prediction

Yuqicheng Zhu[†‡]**, Nico Potyka**[§]**, Mojtaba Nayyeri**[†]**, Bo Xiong**[†]
Yunjie He[†‡]**, Evgeny Kharlamov**[‡♭]**, Steffen Staab**[†♮]
[†]University of Stuttgart, [‡]Bosch Center for AI,
[§]Cardiff University, [♭] University of Oslo, [♮] University of Southampton
yuqicheng.zhu@de.bosch.com

Abstract

Knowledge graph embedding (KGE) models are often used to predict missing links for knowledge graphs (KGs). However, multiple KG embeddings can perform almost equally well for link prediction yet give conflicting predictions for unseen queries. This phenomenon is termed *predictive multiplicity* in the literature. It poses substantial risks for KGE-based applications in high-stake domains but has been overlooked in KGE research. We define predictive multiplicity in link prediction, introduce evaluation metrics and measure predictive multiplicity for representative KGE methods on commonly used benchmark datasets. Our empirical study reveals significant predictive multiplicity in link prediction, with 8% to 39% testing queries exhibiting conflicting predictions. We address this issue by leveraging voting methods from social choice theory, significantly mitigating conflicts by 66% to 78% in our experiments.

1 Introduction

Knowledge graphs (KGs) store factual knowledge of real-world entities and their relationships in the form of triples ⟨*head entity, predicate, tail entity*⟩. KGs allow for logical reasoning and answering of queries. Knowledge graph embeddings (KGE) apply machine-learning methods on KGs to provide extra-logical reasoning capabilities exploiting similarities and analogies over knowledge structures (Ji et al., 2021).

KGE maps entities and predicates into low-dimensional vectors that preserve semantic and structural information of KGs (Hogan et al., 2021). The learned embeddings can be applied to downstream tasks like link prediction. Given queries in the form of ⟨*head entity, predicate, ?*⟩ or ⟨*?, predicate, tail entity*⟩, candidate entities are ranked based on predictive scores provided by KGE models. The positive triples are expected to be ranked higher than the negative triples.
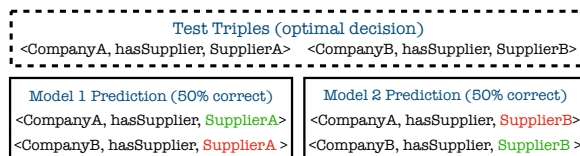

Figure 1: An illustration of predictive multiplicity in link prediction lies within the realm of supplier selection for Company A, where model 1 and 2 are trained with the same KGE algorithm (e.g. TransE) but different random seeds.

The training of the KG embedding introduces randomness into the resulting model. Sources of randomness include randomized parameter initialization, randomized sequences of positive samples, and randomized negative sampling. Given the non-convexity of the training problem, the same KG may lead to various KG embeddings because of the convergence of the training in different local minima. While learned embeddings may exhibit comparable performance in link prediction, they may suggest conflicting predictions for an individual query. This phenomenon is referred to as *predictive multiplicity* in recent literature (Marx et al., 2020; Watson-Daniels et al., 2023; Black et al., 2022b), it is also known as "*Rashomon effect*" and *model multiplicity* in earlier studies (Breiman, 2001). As an example of predictive multiplicity in link prediction, Figure 1 shows the results of two models that both have an overall accuracy of 50%, but predict entirely different facts as top 1 recommendation.

Conflicting predictions introduce considerable risks when applying KGE methods in high-stake domains such as medicine or finance. For example, they would affect treatment decisions, affecting patient health outcomes in the context of medical recommendation (Gong et al., 2021), or switch compounds for confirmatory experiments in drug discovery (Mohamed et al., 2020), potentially altering research direction and efficiency. Moreover, predictive multiplicity complicates the justification

of decisions made from equally accurate models (Black et al., 2022b). For example, when equally accurate models provide contradictory decisions regarding the approval of a loan application (Alam and Ali, 2022), the random selection of a model fails to properly justify the ultimate individual decision. Despite its relevance, predictive multiplicity has been overlooked in KGE research.

To the best of our knowledge, this is the first work to study predictive multiplicity for KGE-based link prediction. Our contribution is two-fold: First, we formally define predictive multiplicity in the context of link prediction. Two metrics, *ambiguity* and *discrepancy*, are introduced to measure predictive multiplicity, with an upper bound derived for discrepancy. Evaluating the predictive multiplicity for six representative KGE methods on commonly used benchmark datasets, we observe significant predictive multiplicity behavior in link prediction, with conflicting predictions ranging from 8% to 39% for testing queries.

To address this issue, our second contribution is to investigate the effectiveness of voting methods from social choice theory in mitigating predictive multiplicity in link prediction. Applying voting methods to aggregate individual rankings yields a more robust ranking that optimizes the collective preference. Our empirical findings demonstrate significant alleviation of predictive multiplicity through voting methods, with the most effective approach reducing conflicting predictions by 66% to 78% for testing queries.

2 Related Work

Although prior studies demonstrate the effectiveness of KGE methods in learning complex patterns in KGs (Bordes et al., 2013; Sun et al., 2019; Nickel et al., 2011; Yang et al., 2015; Trouillon et al., 2016; Dettmers et al., 2018; Xiong et al., 2022b), fewer works focus on their reliability. Some studies address uncertainty quantification in KGE methods (Safavi et al., 2020; Tabacof and Costabello, 2020; Zhu et al., 2024a), and Bhardwaj et al. (2021a,b) explore their adversarial robustness. However, the phenomenon of predictive multiplicity in KGE methods has been overlooked.

The term *model multiplicity* was first discussed in Breiman (2001) with the term "*Rashomon Effect*" referring specifically to the phenomenon where there are different weights learned for linear regression with the same error rate. The term *pre-*

dictive multiplicity was first introduced by Marx et al. (2020), who explored this behavior in binary classification. Marx et al. (2020) further investigate predictive multiplicity in probabilistic classification. Recent studies also provide evidence of predictive multiplicity for deep models (Black et al., 2022a; Mehrer et al., 2020). We initiate an exploration into the predictive multiplicity behavior within the context of KGE-based link prediction.

While predictive multiplicity offers flexibility in model selection without sacrificing accuracy, diverging predictions can result in unjustifiable final choices. Black et al. (2022a) propose a method to provide consistent predictions. Given diverging predictions, they first filter them through a specified confidence threshold and select the final prediction using a majority vote. Besides classification problems, predictive multiplicity is also frequently studied for counterfactual explanations (Pawelczyk et al., 2020; Jiang et al., 2024).

Voting methods can also be seen as ensemble methods. Ensemble strategies are employed in KGE methods (Joshi and Urbani, 2022; Xu et al., 2021) during the training phase to increase the model performance. Joshi and Urbani (2022) focuses on enhancing the accuracy of the triple classification task by aggregating predictions from models trained using different KGE algorithms. Xu et al. (2021) demonstrate that combining multiple low-dimensional models can outperform a single high-dimensional model. However, our approach aggregates rankings using social choice theory in testing time, aiming to alleviate predictive multiplicity by providing more robust rankings.

3 Notations and Preliminaries

3.1 Knowledge Graph Embedding

We consider a KG $\mathcal{G} \subseteq E \times R \times E$ defined over a set E of entities and a set R of relations. The elements in $\mathcal{G}$ are called triples and denoted as $< h, r, t >$. A KGE model $M_\theta : E \times R \times E \to \mathbb{R}$ allocates each triple with a predictive score that measures the plausibility that the triple holds (Bordes et al., 2013). The parameters θ are learned to let M_θ assign higher predictive scores to positive triples (real facts) while assigning lower predictive scores to negative triples (false facts). This can be achieved for example by minimizing *margin-based*

ranking loss (Bordes et al., 2013):

$$\mathcal{L} = \sum_{tr \in \mathcal{T}} \sum_{tr^- \in \mathcal{T}^-} \max(0, \gamma - M_\theta(tr) + M_\theta(tr^-)),$$
(1)

or *cross-entropy loss* (Trouillon et al., 2016):

$$\mathcal{L} = \sum_{tr \in \mathcal{T} \cup \mathcal{T}^-} \log(1 + \exp(-y_{tr} \cdot M_\theta(tr))), \quad (2)$$

where γ is a margin hyperparameter, tr refers to a triple $\langle h, r, t \rangle$, $\mathcal{T}, \mathcal{T}^-$ are the sets of positive and negative triples, respectively. The label of a triple, denoted as y_{tr}, takes values from the set $\{-1, 1\}$. Here, $y_{tr} = 1$ indicates the triple as positive, while $y_{tr} = -1$ indicates that the triple is negative. The negative triples are typically generated by randomly replacing the head entity or the tail entity in a positive triple with a random entity sampled from the entity set.

3.2 Social Choice Theory

Social choice theory studies collective decision-making processes, where individual preferences are aggregated to determine a group's overall preference (Brandt et al., 2016). In this section, we recall some basics of social choice theory from Shoham and Leyton-Brown (2009).

We consider a finite set of candidates $C = \{c_1, \ldots, c_m\}$ and a finite set of voters $V = \{1, \ldots, n\}$, who have different preferences on candidates in C. We represent preferences by a linear order $\succeq$ and let

- $c_1 \succ c_2$ iff $c_1 \succeq c_2 \wedge c_2 \not\succeq c_1$ (*strict preference*)

- $c_1 \sim c_2$ iff $c_1 \succeq c_2 \wedge c_2 \succeq c_1$ (*indifference*)

We let $\succeq_i$ denote the preference ordering of the i-th voter. A *preference profile* $p : [\succeq_1, \ldots, \succeq_n]$ is a list of preference orderings. Next, we introduce some interesting voting methods from social choice theory (Brandt et al., 2016).

Definition 1 (Scoring Rule). *A score vector is a vector $\mathbf{w} \in \mathbb{R}^m$ such that $w_1 \geq w_2 \geq \cdots \geq w_m$ and $w_1 > w_m$. Any score vector induces a scoring rule, in which each voter awards w_1 points to the top-ranked candidate, w_2 points to the second-ranked, and so on. The candidate with the highest total sum of scores wins.*

Definition 2 (Majority Voting). *Majority voting is a scoring rule with the score vector $(1, 0, \ldots, 0)$.*

Definition 3 (Borda Voting). *Given m candidates, Borda voting is a scoring rule with the score vector $(m - 1, m - 2, \ldots, 0)$.*

Definition 4 (Range Voting (Smith, 2000)). *Given m candidates, range voting is a scoring rule with a score vector $\mathbf{w} \in [-1, 1]^m$.*

Additionally, we introduce several properties desirable for the link prediction task in Appendix A.

4 Predictive Multiplicity in Link Prediction

4.1 Link Prediction

A query $q \in Q$ is of the form $\langle h, r, ? \rangle$ or $\langle ?, r, t \rangle$. We let $tr(q, e)$ denote the corresponding triple $\langle h, r, e \rangle$ or $\langle e, r, t \rangle$, respectively. A KGE model M_θ can be used to rank the candidate entities for query q. We define the ranking $\succeq_{M_\theta, q}$ by $e_1 \succeq_{M_\theta, q} e_2$ iff $M_{\theta, q}(tr(q, e_1)) \geq M_{\theta, q}(tr(q, e_2))$. We let $R_{\succeq_{M_\theta, q}}(e)$ denote the rank position of a specific candidate entity $e \in E$, that is

$$R_{\succeq_{M_\theta, q}}(e) = 1 + |\{d \in E \mid d \succeq_{M_\theta, q} e\}| \quad (3)$$

Then the link prediction task can be formulated as a binary classification problem: determine whether a triple is ranked within the top-K predictions:

$$T_K(M_\theta, tr(q, e)) = \mathbb{1}[R_{\succeq_{M_\theta, q}}(e) \leq K]. \quad (4)$$

The performance of link prediction is commonly evaluated by $Hits@K$. The test set $\mathcal{T}$ contains testing queries (q, e) consisting of a query q and a correct answer e. We define the $Hits@K$ function H_K of a KGE model M_θ as

$$H_K(M_\theta) = \frac{1}{|\mathcal{T}|} \sum_{(q,e) \in \mathcal{T}} \mathbb{1}[R_{\succeq_{M_\theta, q}}(e) \leq K] \quad (5)$$

4.2 Definition of Predictive Multiplicity

We study KGE models that perform similarly in link prediction task in terms of $Hits@K$, i.e. competing models. Following Marx et al. (2020), we will now define a ϵ-*level set* for similar performing models and ϵ as the *error tolerance*.

We let $\mathcal{M}$ denote a hypothesis class of KGE models. A *baseline model* $M_\theta^* \in \mathcal{M}$ is the KGE model that achieves the highest $Hits@K$ on the validation dataset throughout the hyperparameter optimization process. $D(M_\theta, M_\theta^*)$ measures the difference between baseline model and a competing model with respect to Hits@K.

$$D(M_\theta, M_\theta^*) = H_K(M_\theta^*) - H_K(M_\theta). \quad (6)$$

336

Definition 5 (ϵ-level set). *Given a baseline KGE model M_θ^* and a hypothesis class $\mathcal{M}$, the ϵ-level set around M_θ^* is the set of all models $M_\theta \in \mathcal{M}$ with a performance difference at most ϵ in the link prediction task.*

$$S_\epsilon(M_\theta^*) := \{M_\theta \in \mathcal{M} \mid D(M_\theta, M_\theta^*) \leq \epsilon\}, \quad (7)$$

Given a testing query set $\mathcal{T}$, predictive multiplicity is defined for testing queries $\tau = (q, e)$ that receive conflicting predictions from competing models.

Definition 6 (Predictive Multiplicity). *Given a baseline KGE model M_θ^*, an error tolerance ϵ and a testing query set $\mathcal{T}$, link prediction problem exhibits predictive multiplicity over the ϵ-level set $S_\epsilon(M_\theta^*)$ if there exists a model $M_\theta \in S_\epsilon(M_\theta^*)$ such that $T_K(M_\theta, tr(\tau_i)) \neq T_K(M_\theta^*, tr(\tau_i))$ for some $\tau_i \in \mathcal{T}$.*

4.3 Measuring Predictive Multiplicity

Ambiguity and *discrepancy* are two measures that have been used to quantify predictive multiplicity in classification tasks (Marx et al., 2020; Watson-Daniels et al., 2023). We next define them for link prediction.

To make the notation more concise, we use $\Delta(M_\theta, \tau)$ to denote whether a competing model M_θ provides conflicting predictions compared to the baseline model M_θ^* for a testing query $\tau = (q, e)$.

$$\Delta(M_\theta, \tau) = \mathbb{1}[T_K(M_\theta, tr(\tau)) \neq T_K(M_\theta^*, tr(\tau))] \quad (8)$$

Definition 7 (Ambiguity). *Given a testing query set $\mathcal{T}$, the ambiguity of link prediction over the ϵ-level set $S_\epsilon(M_\theta^*)$ is the proportion of testing queries that obtain a different prediction by a competing model $M_\theta \in S_\epsilon(M_\theta^*)$:*

$$\alpha_\epsilon(M_\theta^*) := \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \max_{M_\theta \in S_\epsilon(M_\theta^*)} \Delta(M_\theta, \tau) \quad (9)$$

Definition 8 (Discrepancy). *The discrepancy of link prediction over the ϵ-level set $S_\epsilon(M_\theta^*)$ is the maximum percentual disagreement between the baseline model and a competing model $M_\theta \in S_\epsilon(M_\theta^*)$:*

$$\delta_\epsilon(M_\theta^*) := \max_{M_\theta \in S_\epsilon(M_\theta^*)} \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \Delta(M_\theta, \tau) \quad (10)$$

Ambiguity measures the proportion of testing queries that exhibit predictive multiplicity, while discrepancy captures the largest fraction of test queries for which the predicted answers vary upon switching the baseline model with a competing model.

4.4 Bound on Predictive Multiplicity

In Proposition 1, we bound the number of queries with conflicting predictions between the baseline model and a competing model in the ϵ-level set. We provide a proof in Appendix B.

Proposition 1 (Bound on Discrepancy). *The discrepancy between the baseline model M_θ^* and any competing model $M_\theta \in S_\epsilon(M_\theta^*)$ obeys:*

$$\delta_\epsilon(M_\theta^*) \leq 2 \cdot (1 - H_K(M_\theta^*)) + \epsilon \quad (11)$$

The upper bound illustrates how the extent of predictive multiplicity depends on $Hits@K$ of the baseline model. Specifically, a less accurate baseline model theoretically provides greater potential for predictive multiplicity.

5 Alleviating Predictive Multiplicity using Social Choice Theory

The predictive multiplicity can be alleviated by improving the robustness of the rankings. Here, robustness means models with similar performance should also provide similar rankings for testing queries. Social choice theory provides a theoretical framework for aggregating individual preferences to determine a group's overall preference (Brandt et al., 2016). Voting methods from social choice theory can help "smooth out" the randomness in rankings by aggregating individual models (Potyka et al., 2024). Intuitively, the candidate entities that are constantly ranked high for all models should also be ranked high in final rankings.

We next describe ranking aggregation using voting methods with a running example and adapt range voting (Smith, 2000) to aggregate the predictive scores for the final ranking.

5.1 Ranking Aggregation using Voting Methods

For link prediction, given a query q and a KGE model M_θ, the ranking of candidate entities for a query is denoted as $\succeq_{M_\theta, q}$. By training KGE models with N different random seeds, we obtain a profile for each query $p_q = [\succeq_{M_\theta, q}^1, \ldots, \succeq_{M_\theta, q}^N]$. A ranking aggregation process takes p_q as input and outputs a single ranking.

We illustrate how to aggregate rankings with voting methods in link prediction task with the following running example.

Example 1. *Assume there are in total four entities* $\{A, B, C, D\}$ *and one relation* r *in our KG. Given a query* $\langle A, r, ? \rangle$*, three models* $[M_\theta^1, M_\theta^2, M_\theta^3]$ *sampled from* ϵ*-level set* $S_\epsilon(M_\theta^*)$ *provide different rankings in Table 1. The predictive scores for candidate entities is shown in brackets after each entity.*

Model ID	Rankings
1	$C(100) \succ_1 B(8) \succ_1 D(6) \succ_1 A(1)$
2	$B(8) \succ_2 D(7) \succ_2 C(6) \succ_2 A(5)$
3	$B(40) \succ_3 C(10) \succ_3 A(2) \succ_3 D(1)$

Table 1: Rankings of models with corresponding predictive scores for query $\langle A, r, ? \rangle$.

We apply all voting methods described in section 3.2 for ranking aggregation. Majority voting and Borda voting aggregate rankings are based on ordinal positions of candidates, while range voting assigns more informative scores to candidates. To adapt range voting in link prediction, we transform the predictive scores into scores within range $[-1, 1]$. Concretely, we denote the predictive scores of candidate entities for a query as $\Gamma = [\gamma_1, \ldots, \gamma_{|E|}]$ and the score vector of range voting as $\mathbf{w} = [w_1, \ldots, w_{|E|}]$. We then obtain the score vector based on predictive scores as follows:

$$w_i = 2 \times \frac{\gamma_i - \min(\Gamma)}{\max(\Gamma) - \min(\Gamma)} - 1. \quad (12)$$

Table 2 shows the scores assigned to candidate entities by voting methods, which are then used to re-rank the entities based on the sum of their voting scores. The resulting aggregated rankings are presented in Table 3.

Entity	Majority Vote				Borda Vote				Range Vote			
	$\succ_1$	$\succ_2$	$\succ_3$	sum	$\succ_1$	$\succ_2$	$\succ_3$	sum	$\succ_1$	$\succ_2$	$\succ_3$	sum
A	0	0	0	0	0	0	1	1	-1	-1	-0.95	-2.95
B	0	1	1	2	2	3	3	8	-0.85	1	1	1.15
C	1	0	0	1	3	1	2	6	1	0.33	-0.54	0.79
D	0	0	0	0	1	2	0	3	-0.90	-0.33	-1	-2.23

Table 2: Ranking aggregation process for Example 1.

6 Experiments

In this section, we measure the predictive multiplicity in link prediction and apply voting methods

Voting Method	Rankings
Majority Vote	$B(2) \succ C(1) \succ D(0) \sim A(0)$
Borda Vote	$B(8) \succ C(6) \succ D(3) \succ A(1)$
Range Vote	$B(1.15) \succ C(0.79) \succ D(-2.23) \succ A(-2.95)$

Table 3: Aggregated rankings of different voting methods for Example 1.

from social choice theory. Our goals are (i) to measure the predictive multiplicity for the link prediction task; (ii) to investigate to which extent voting methods can alleviate predictive multiplicity.

Models and Datasets. The main experiments are conducted for six representative KGE models (TransE (Bordes et al., 2013), RotatE (Sun et al., 2019), RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), and ConvE (Dettmers et al., 2018)) on four public benchmark datasets (WN18 (Bordes et al., 2013), WN18RR (Dettmers et al., 2018), FB15k (Bordes et al., 2013), and FB15k-237 (Toutanova and Chen, 2015)). A small dataset Nations (Hoyt et al., 2022) is additionally used for investigating the change of predictive multiplicity with respect to the error tolerance ϵ. The statistics of benchmark datasets are summarized in Table 4.

	#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
Nations	14	55	1,592	199	201

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

Experiment Settings. For training KGE, we use the implementation of LibKGE (Broscheit et al., 2020). All experiments were conducted on a Linux machine with a 40GB NVIDIA A100 SXM4 GPU.

6.1 Evaluating Predictive Multiplicity

The ϵ-level set, as defined in Definition 5, is too large to be evaluated in practice. As usual, we will use empirical notions of ambiguity and discrepancy that are based on a sample of the ϵ-level set that we denote by $S_\epsilon(M_\theta^*)'$.

Constructing the Subset of ϵ-level Set. To construct $S_\epsilon(M_\theta^*)'$, we first obtain the baseline model M_θ^* by performing 60 trials of hyperparameter search using the strategy in Ruffinelli et al. (2019) (more details in Appendix C) and set ϵ to 0.01 (a commonly used value in the literature (Marx et al., 2020; Watson-Daniels et al., 2023)). Subsequently,

we train a potential competing model using the training configurations of the baseline model with a different random seed. If the performance difference between the potential competing model and the baseline model is less than ϵ, we add it in $S_\epsilon(M_\theta^*)'$. Due to computational constraints, we limit the size of $S_\epsilon(M_\theta^*)'$ to 10 in our experiment. Refer to Algorithm 1 for a pseudocode outlining this process.

Algorithm 1 Pseudocode for $S_\epsilon(M_\theta^*)'$ construction.

1: $M_\theta^* \leftarrow$ Bayesian Optimization for 60 trials.
2: $\epsilon \leftarrow 0.01$.
3:
4: $S_\epsilon(M_\theta^*)' \leftarrow$ An empty set.
5: **while** $|S_\epsilon(M_\theta^*)'| \leq 10$ **do**
6:     $M_\theta \leftarrow$ Retrain M_θ^* with a different random seed.
7:     **if** $D(M_\theta, M_\theta^*) \leq \epsilon$ **then**
8:         $S_\epsilon(M_\theta^*)'$ add M_θ.
9:     **end if**
10: **end while**
11: **return** $S_\epsilon(M_\theta^*)'$

Evaluation Metrics. We evaluate the accuracy of link prediction with $Hits@K$ and the predictive multiplicity with ambiguity and discrepancy. Note that in our experiment, ambiguity and discrepancy are measured by their empirical counterpart over the ϵ-level set approximation $S_\epsilon(M_\theta^*)'$. To distinguish these metrics from previous definitions in section 4.3, we denote them as $\hat{\alpha}_\epsilon$ and $\hat{\delta}_\epsilon$ and call them empirical ambiguity and discrepancy, respectively.

Evaluation Procedure. We demonstrate the evaluation procedure in Algorithm 2. We denote $Aggregate(A, S_{agg})$ as a procedure to aggregate rankings predicted by models in S_{agg} using a voting method A (detailed in section 5.1). The result of $Aggregate(A, S_{agg})$ can be viewed as a new KGE model M_{agg} that predicts the aggregated rankings. train(config(M), $seed$) denotes the training process of a KGE model, which adopts the same training configurations (including the training graph, hyperparameters, etc.) of a pre-trained model M with a specific random seed.

For each KGE method and benchmark dataset, we first construct a set of competing models, $S_\epsilon(M_\theta^*)'$. Without employing voting methods, we assess $Hits@K$, $\hat{\alpha}_\epsilon$, and $\hat{\delta}_\epsilon$ over $S_\epsilon(M_\theta^*)'$. Oth-

---

Algorithm 2 Pseudocode for evaluation.

Require: $S_\epsilon(M_\theta^*)'$
1: $S \leftarrow$ An empty set. ▷ Initialize evaluation set
2: **if** not apply voting method A **then**
3:     $S \leftarrow S_\epsilon(M_\theta^*)'$
4: **else**
5:     **for each** M_θ in $S_\epsilon(M_\theta^*)'$ **do**
6:         $S_{agg} \leftarrow$ An empty set.
7:         **for** $i \leftarrow 1$ to 10 **do**
8:             $seed_i \leftarrow$ generateRandomSeed()
9:             $\hat{M}_\theta \leftarrow$ train(config(M_θ), $seed_i$).
10:             $S_{agg} \leftarrow S_{agg} \cup \{\hat{M}_\theta\}$
11:         **end for**
12:         $M_{agg} \leftarrow$ Aggregate(A, S_{agg}).
13:         $S \leftarrow S \cup \{M_{agg}\}$.
14:     **end for**
15: **end if**
16:
17: Evaluate $Hits@K$ for all models in S and report the average value.
18: Evaluate $\hat{\alpha}_\epsilon$ and $\hat{\delta}_\epsilon$ for S.

---

erwise, we collect a set of models S_{agg} for each model M_θ in $S_\epsilon(M_\theta^*)'$ by training 10 models using the configurations of M_θ with different random seeds. Subsequently, we aggregate the models in S_{agg} with a voting method A to get an "aggregated" model M_{agg} for each M_θ, and then measure all metrics over the set of aggregated models.

Results. We present the results of predictive multiplicity of link prediction in Table 5. For benchmark datasets WN18RR, FB15k237 and six KGE representative methods, we observe that competing models with less than 1% error tolerance ($\epsilon = 0.01$) assign conflicting predictions for 8% to 39% of testing queries ($\hat{\alpha}_\epsilon$). Voting methods effectively mitigate the issue of predictive multiplicity. Majority voting generally reduces conflicting predictions but also decreases $Hit@K$ substantially. Borda voting yields comparable $Hits@K$ and significantly alleviate predictive multiplicity. Range voting consistently outperforms other methods in terms of $Hits@K$ and substantially reduces predictive multiplicity, resulting in a relative decrease of 66% to 78% in empirical ambiguity ($\hat{\alpha}_\epsilon$) and 64% to 76% in empirical discrepancy ($\hat{\delta}_\epsilon$).

We focus on link prediction for recommendation, emphasizing the importance of whether true facts are ranked within the top-K. In Appendix D.2, we extend our analysis to link prediction within a query answering context, where the objective is to

Models	Baselines	WN18RR			FB15k237		
		$Hits@10\uparrow$	$\hat{\alpha}_\epsilon\downarrow$	$\hat{\delta}_\epsilon\downarrow$	$Hits@10\uparrow$	$\hat{\alpha}_\epsilon\downarrow$	$\hat{\delta}_\epsilon\downarrow$
TransE	w/o	0.518	0.076	0.034	0.455	0.385	0.145
	major	0.055	0.096	0.045	0.155	0.171	0.081
	Borda	0.482	0.032	0.016	0.456	0.110	0.044
	range	0.519	**0.017**	**0.009**	0.470	0.101	0.041
RotatE	w/o	0.547	0.195	0.074	0.520	0.163	0.064
	major	0.413	0.064	0.029	0.204	0.104	0.053
	Borda	0.564	0.062	0.028	0.523	0.039	0.017
	range	**0.578**	0.051	0.022	**0.524**	**0.037**	**0.016**
RESCAL	w/o	0.517	0.248	0.095	0.482	0.375	0.140
	major	0.198	0.108	0.054	0.145	0.165	0.089
	Borda	0.561	0.099	0.043	0.485	0.107	0.048
	range	0.575	0.084	0.034	0.498	0.098	0.042
DistMult	w/o	0.526	0.169	0.068	0.476	0.320	0.120
	major	0.185	0.078	0.037	0.144	0.124	0.059
	Borda	0.524	0.055	0.024	0.475	0.088	0.037
	range	0.542	0.048	0.021	0.488	0.082	0.034
ComplEx	w/o	0.541	0.217	0.085	0.482	0.308	0.116
	major	0.243	0.243	0.126	0.145	0.121	0.055
	Borda	0.559	0.067	0.030	0.480	0.087	0.036
	range	0.573	0.058	0.024	0.493	0.082	0.032
ConvE	w/o	0.500	0.222	0.088	0.474	0.340	0.130
	major	0.185	0.092	0.047	0.150	0.154	0.074
	Borda	0.522	0.082	0.035	0.474	0.092	0.039
	range	0.534	0.068	0.027	0.486	0.085	0.034

Table 5: This table compares the accuracy and predictive multiplicity of applying different voting methods on six representative KGE models and two benchmark datasets, WN18RR and FB15k237. We underline the best values for each model-dataset pair and boldface the global optimal values. (Results for more datasets see Table 7 in Appendix D.1.)

Figure 2: Predictive multiplicity for ComplEx on Nations dataset wrt. ϵ.

6.2.2 Investigating the Number of Models for Aggregation

In Figure 3, we investigate the predictive multiplicity metrics in relation to the number of models employed for ranking aggregation. Employing a larger number of models for aggregation yields a more notable alleviation of predictive multiplicity. Remarkably, even with a relatively small number of aggregated models, substantial improvements in predictive multiplicity can be attained. Furthermore, change of the number of models for aggregation does not notably affect $Hits@K$ (Figure 11 - 14 in Appendix D.4).
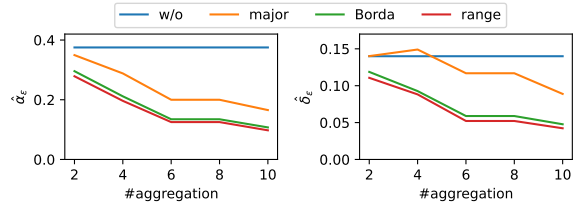
Figure 3: Investigation of the predictive multiplicity with respect to the number of models used for voting methods. Due to page limit, we only show the results of RESCAL on FB15k237 in this figure, we put more results in appendix D.4.

6.2.3 Investigating the Predictive Multiplicity wrt. Entity/Relation Frequency

Most entities/relations only have a few facts in KGs (Xiong et al., 2018). There are more possible embeddings or more uncertainty for those relations/entities since they are less constrained by the existing facts in KG during training. Intuitively, there might be more significant predictive multiplicity behavior for queries containing those entities/relations.

We conduct hypothesis tests using Spearman's coefficient (ρ) to assess the correlation between entity/relation frequency (i.e., the number of triples

determine whether competing models yield similar/same answer sets. Comparable conclusions can be drawn within that context as well.

6.2 Further Analysis

6.2.1 Investigating Predictive Multiplicity wrt. Error Tolerance

We conduct experiment for ComplEx on Nations to investigate the influence of ϵ on predictive multiplicity. The procedure follows Algorithm 2 with thirty values of ϵ spanning the range from 0 to 0.06. We represent the results in Figure 2. Our observations confirm the expectation in section 4.4: both predictive multiplicity metrics increase with larger values of ϵ. Employing voting methods consistently reduces both ambiguity and discrepancy across all ϵ values, with a more pronounced effect observed for larger ϵ. Notably, even at $\epsilon = 0$, conflicting predictions persist, underscoring the necessity to report predictive multiplicity even for equally accurate models. Additionally, we observe that the change of ϵ has negligible effects on $Hits@K$, as detailed in Appendix D.3.
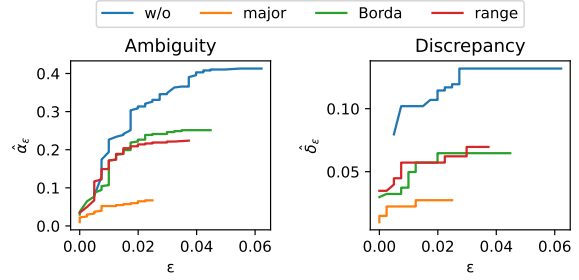
		w/o		range vote	
Var.1	Var.2	ρ	p-value	ρ	p-value
Rel. Fre	$\hat{\alpha}_\epsilon$	-0.349	<0.001	-0.156	<0.001
Rel. Fre	$\hat{\delta}_\epsilon$	-0.400	<0.001	-0.204	<0.001
Ent. Fre	$\hat{\alpha}_\epsilon$	-0.106	<0.001	-0.098	<0.001
Ent. Fre	$\hat{\delta}_\epsilon$	-0.114	<0.001	-0.103	<0.001

Table 6: This table presents the correlation between entity/relation frequency and $\hat{\alpha}_\epsilon$ and $\hat{\delta}_\epsilon$, with Spearman's coefficient (ρ) and its p-value. Columns 3 and 4 show results without applying voting method, while columns 5 and 6 show results with range voting.

containing the target entity/relation) and predictive multiplicity metrics ($\hat{\alpha}_\epsilon$ and $\hat{\delta}_\epsilon$). ρ ranges from -1 to 1, indicating the strength and direction of the correlation: close to 1 implies a positive monotonic relationship, while close to -1 implies a monotonic negative relationship.

We count entity/relation frequencies (Ent. Fre and Rel. Fre) as variable 1 and calculate $\hat{\alpha}_\epsilon$ and $\hat{\delta}_\epsilon$ for six KGE methods on entity/relation- specific subsets of all datasets as variable 2. Results in Table 6 show a significant negative correlation, confirming our conjecture. Notably, applying range voting weakens this correlation, potentially due to its effectiveness in alleviating predictive multiplicity for queries with higher uncertainty.

7 Discussing Other Influential Factors of Predictive Multiplicity

In this section, we discuss additional factors that may influence predictive multiplicity, namely expressiveness and inference patterns. We briefly introduce these two notions and then discuss some observations regarding their relationship to predictive multiplicity.

Expressiveness. The expressiveness of KGE models refers to the ability of modeling an arbitrary KG. Following (Pavlović and Sallinger, 2023; Wang et al., 2018), we call a KGE model fully expressive if we can find a parameter set such that the model predicts all training triples correctly. Intuitively, more expressive models can represent more possible embeddings that fit the training graph, thereby allowing more "room" for multiplicity.

Inference Patterns. Inference patterns refer to the logic rules used to derive new triples from the observed facts in KGs. The generalization capabilities of KGE is usually analysed based on inference patterns that KGE model can capture (Abboud et al., 2020). For instance, TransE can capture inverse

patterns, wherein $r_1(X, Y)$ implies $r_2(Y, X)$, suggesting that the testing triple $\langle e_1, r_2, e_2 \rangle$ can be correctly predicted with low uncertainty if $\langle e_2, r_1, e_1 \rangle$ is present in the training graph. Theoretically, if the KGE method effectively captures the inference patterns for the testing triple, we would expect fewer conflicts from competing models.

Observations. According to Wang et al. (2018, Table 1), RESCAL and ComplEx are more expressive than DistMult when considering similar embedding dimensions. We observe that RESCAL and ComplEx associate with larger values of ambiguity and discrepancy than DistMult in Table 5, aligning with our conjecture regarding expressiveness. Furthermore, WN18 and FB15k are known to suffer from test leakage due to inverse relations (Toutanova and Chen, 2015), meaning that many test triples can be easily derived by the inverse pattern. WN18RR and FB15k-237 delete inverse relations to address this issue (Toutanova and Chen, 2015; Dettmers et al., 2018). In Figure 4, we note a consistent trend where competing models exhibit fewer conflicting predictions on WN18 and FB15k compared to WN18RR and FB15k237. This observation supports our conjecture regarding inference patterns, as the absence of even a single inference pattern notably increase the number of conflicting predictions.
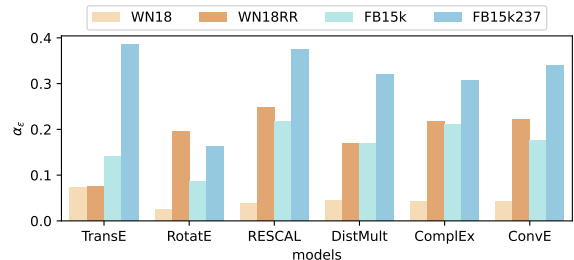

Figure 4: We demonstrate the ambiguity for 10 competing models on WN18, WN18RR, FB15k and FB15k237 in this figure.

The result of TransE on FB15k237 appears to be an outlier, marked by its low expressiveness but the highest ambiguity and discrepancy among all KGE methods. However, in FB15k237, numerous training triples involve symmetric relations, with testing triples inferrable through symmetric patterns (Rim et al., 2021). Since TransE fails to represent symmetric triple pair ($\langle e_1, r, e_2 \rangle$ and $\langle e_2, r, e_1 \rangle$) simultaneously and lacks the capability to capture symmetric patterns, it may therefore exhibit additional predictive multiplicity.

8 Conclusion

In this paper, we define and measure the predictive multiplicity in link prediction. We measure the predictive multiplicity with empirical ambiguity and discrepancy for representative KGE methods on commonly used benchmark datasets. Our empirical study reveals significant predictive multiplicity in link prediction, and we demonstrate the effectiveness of applying voting methods. We also discuss several potential factors that could influence predictive multiplicity in link prediction.

Furthermore, according to Proposition 1, predictive multiplicity depends on the accuracy of the baseline model and error tolerance (ϵ). A less accurate baseline model or larger ϵ allows for more predictive multiplicity. Given the typically low accuracy in link prediction and the existence of conflicting predictions even when $\epsilon = 0$, a considerable number of conflicting predictions may arise from competing models in practice, posing significant risks in safety-critical domains. Hence, we advocate for the measurement, reporting, and mitigation of predictive multiplicity in link prediction within these domains.

9 Limitations

In Section 7, we offer conjectures regarding the relationship between influential factors and predictive multiplicity. Our findings only show that our conjectures are potentially reasonable, but no conclusions can be drawn based on them. A systematic analysis necessitates quantifying expressiveness, inference patterns, which falls outside the scope of our paper but is a promising avenue for future research.

To mitigate predictive multiplicity, employing voting methods derived from social choice theory emerges as a straightforward yet effective strategy. However, voting-based ranking aggregation requires training multiple competing models from scratch, which can be time/computational consuming. Addressing predictive multiplicity during the training phase is considered as next step. Furthermore, more advanced voting methods such as partial Borda voting (Cullinan et al., 2014) could be explored in the future, which aggregates only partial rankings to reduce memory requirements during the aggregation step.

10 Ethics Statement

In this study, we emphasize the importance of reporting and dealing with predictive multiplicity to ensure fair and transparent decision-making processes for KGE-based applications. Failure to account for predictive multiplicity may lead decision-makers to select models that align with their personal preferences, potentially resulting in unfair outcomes for individuals. By neglecting to report predictive multiplicity of KGE models, decision-makers risk undermining the integrity and equity of the decision-making process.

References

Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. Boxe: A box embedding model for knowledge base completion. In *NeurIPS*.

Md. Nurul Alam and Muhammad Masroor Ali. 2022. Loan default risk prediction using knowledge graph. In *KST*, pages 34–39. IEEE.

Erik Arakelyan, Daniel Daza, Pasquale Minervini, and Michael Cochez. 2021. Complex query answering with neural link predictors. In *ICLR*. OpenReview.net.

Kenneth J. Arrow. 1951. *Social Choice and Individual Values*.

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

Peru Bhardwaj, John D. Kelleher, Luca Costabello, and Declan O'Sullivan. 2021a. Adversarial attacks on knowledge graph embeddings via instance attribution methods. In *EMNLP (1)*, pages 8225–8239. Association for Computational Linguistics.

Peru Bhardwaj, John D. Kelleher, Luca Costabello, and Declan O'Sullivan. 2021b. Poisoning knowledge graph embeddings via relation inference patterns. In *ACL/IJCNLP (1)*, pages 1875–1888. Association for Computational Linguistics.

Emily Black, Klas Leino, and Matt Fredrikson. 2022a. Selective ensembles for consistent predictions. In *ICLR*. OpenReview.net.

Emily Black, Manish Raghavan, and Solon Barocas. 2022b. Model multiplicity: Opportunities, concerns, and solutions. In *FAccT*, pages 850–863. ACM.

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.

Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors. 2016. *Handbook of Computational Social Choice*. Cambridge University Press.

Leo Breiman. 2001. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231.

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.

Kewei Cheng, Jiahao Liu, Wei Wang, and Yizhou Sun. 2022. Rlogic: Recursive logical rule learning from knowledge graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 179–189.

John Cullinan, Samuel K Hsiao, and David Polett. 2014. A borda count for partially ordered ballots. *Social Choice and Welfare*, 42:913–926.

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.

Zifeng Ding, Yifeng Li, Yuan He, Antonio Norelli, Jingcheng Wu, Volker Tresp, Yunpu Ma, and Michael Bronstein. 2024a. Dygmamba: Efficiently modeling long-term temporal dependency on continuous-time dynamic graphs with state space models. *arXiv preprint arXiv:2408.04713*.

Zifeng Ding, Jingcheng Wu, Jingpei Wu, Yan Xia, and Volker Tresp. 2024b. Temporal fact reasoning over hyper-relational knowledge graphs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Fan Gong, Meng Wang, Haofen Wang, Sen Wang, and Mengyue Liu. 2021. Smr: medical knowledge graph embedding for safe medicine recommendation. *Big Data Research*, 23:100174.

Yunjie He, Daniel Hernandez, Mojtaba Nayyeri, Bo Xiong, Yuqicheng Zhu, Evgeny Kharlamov, and Steffen Staab. 2024. Generating $\mathcal{SROI}^-$ ontologies via knowledge graph query embedding learning. In *ECAI*. IOS Press.

Yunjie He, Mojtaba Nayyeri, Bo Xiong, Yuqicheng Zhu, Evgeny Kharlamov, and Steffen Staab. 2023. Can pattern learning enhance complex logical query answering? In *ISWC (Posters/Demos/Industry)*, volume 3632 of *CEUR Workshop Proceedings*. CEUR-WS.org.

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.

Charles Tapley Hoyt, Max Berrendorf, Mikhail Galkin, Volker Tresp, and Benjamin M Gyori. 2022. A unified framework for rank-based evaluation metrics for link prediction in knowledge graphs. *arXiv preprint arXiv:2203.07544*.

Paul Jaccard. 1901. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bull Soc Vaudoise Sci Nat*, 37:241–272.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514.

Junqi Jiang, Francesco Leofante, Antonio Rago, and Francesca Toni. 2024. Recourse under model multiplicity via argumentative ensembling. In *AAMAS*, pages 954–963. International Foundation for Autonomous Agents and Multiagent Systems / ACM.

Unmesh Joshi and Jacopo Urbani. 2022. Ensemble-based fact classification with knowledge graph embeddings. In *European Semantic Web Conference*, pages 147–164. Springer.

Charles T. Marx, Flávio P. Calmon, and Berk Ustun. 2020. Predictive multiplicity in classification. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 6765–6774. PMLR.

Johannes Mehrer, Courtney J Spoerer, Nikolaus Kriegeskorte, and Tim C Kietzmann. 2020. Individual differences among deep neural network models. *Nature communications*, 11(1):5725.

Sameh K Mohamed, Vít Nováček, and Aayah Nounu. 2020. Discovering protein drug targets using knowledge graph embeddings. *Bioinformatics*, 36(2):603–610.

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.

Aleksandar Pavlović and Emanuel Sallinger. 2023. Expressive: A spatio-functional embedding for knowledge graph completion. In *The Eleventh International Conference on Learning Representations*.

Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. 2020. On counterfactual explanations under predictive multiplicity. In *Conference on Uncertainty in Artificial Intelligence*, pages 809–818. PMLR.

Nico Potyka. 2018. Continuous dynamical systems for weighted bipolar argumentation. In *KR*, pages 148–157. AAAI Press.

Nico Potyka. 2021. Interpreting neural networks as quantitative argumentation frameworks. In *AAAI*, pages 6463–6470. AAAI Press.

Nico Potyka, Yuqicheng Zhu, Yunjie He, Evgeny Kharlamov, and Steffen Staab. 2024. Robust knowledge extraction from large language models using social choice theory. In *AAMAS*, pages 1593–1601. International Foundation for Autonomous Agents and Multiagent Systems / ACM.

Meng Qu and Jian Tang. 2019. Probabilistic logic neural networks for reasoning. *Advances in neural information processing systems*, 32.

Wiem Ben Rim, Carolin Lawrence, Kiril Gashteovski, Mathias Niepert, and Naoaki Okazaki. 2021. Behavioral testing of knowledge graph embedding models for link prediction. In *3rd Conference on Automated Knowledge Base Construction*.

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.

Yoav Shoham and Kevin Leyton-Brown. 2009. *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.

Warren D Smith. 2000. Range voting. *The paper can be downloaded from the author's homepage at http://www. math. temple. edu/~ wds/homepage/works. html*.

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.

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.

Sergei Vasiljev. 2014. Cardinal voting: the way to escape the social choice impossibility. *Vasiljev SA CARDINAL VOTING: THE WAY TO ESCAPE THE SOCIAL CHOICE IMPOSSIBILITY. Young Scientist USA. Social science. Auburn, USA*, pages 80–85.

Yanjie Wang, Rainer Gemulla, and Hui Li. 2018. On multi-relational link prediction with bilinear models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jamelle Watson-Daniels, David C. Parkes, and Berk Ustun. 2023. Predictive multiplicity in probabilistic classification. In *AAAI*, pages 10306–10314. AAAI Press.

Bo Xiong, Mojtaba Nayyeri, Linhao Luo, Zihao Wang, Shirui Pan, and Steffen Staab. 2024. Neste: Modeling nested relational structures for knowledge graph reasoning. *The 38th Annual AAAI Conference on Artificial Intelligence*.

Bo Xiong, Mojtaba Nayyeri, Shirui Pan, and Steffen Staab. 2023. Shrinking embeddings for hyper-relational knowledge graphs. *The 61st Annual Meeting of the Association for Computational Linguistics*.

Bo Xiong, Nico Potyka, Trung-Kien Tran, Mojtaba Nayyeri, and Steffen Staab. 2022a. Faithful embeddings for el++ knowledge bases. In *ISWC*, volume 13489 of *Lecture Notes in Computer Science*, pages 22–38. Springer.

Bo Xiong, Shichao Zhu, Mojtaba Nayyeri, Chengjin Xu, Shirui Pan, Chuan Zhou, and Steffen Staab. 2022b. Ultrahyperbolic knowledge graph embeddings. In *KDD*, pages 2130–2139. ACM.

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.

Chengjin Xu, Mojtaba Nayyeri, Sahar Vahdati, and Jens Lehmann. 2021. Multiple run ensemble learning with low-dimensional knowledge graph embeddings. In *IJCNN*, pages 1–8. IEEE.

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)*.

H Peyton Young. 1975. Social choice scoring functions. *SIAM Journal on Applied Mathematics*, 28(4):824–838.

Yuqicheng Zhu, Nico Potyka, Jiarong Pan, Bo Xiong, Yunjie He, Evgeny Kharlamov, and Steffen Staab. 2024a. Conformalized answer set prediction for knowledge graph embedding. *arXiv preprint arXiv:2408.08248*.

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 Properties of Voting Methods from Social Choice Theory

All voting methods were proposed to aggregated preferences in an intuitive "fair" way. However, for some cases, they may fail unintendedly. Thus, precisely defined properties - appealing behaviors that the voting methods satisfy, are investigated in social choice theory (Brandt et al., 2016).

We introduce some properties from (Brandt et al., 2016) that are desirable for link prediction. Recall that a *social choice function* is a function f mapping from the set of all possible profiles $\mathcal{P}$ to a non-empty subset of possible candidate C. Given a finite set of voters $N = \{1, \ldots, n\}$ and a profile $p = [\succeq_1, \ldots, \succeq_n]$, f is called:

- **anonymous**: if f does not depend on the identity voters, i.e., if for every bijective function $\pi : V \to V$, we have $f([\succeq_1, \ldots, \succeq_n]) = f([\succeq_{\pi(1)}, \ldots, \succeq_{\pi(n)}])$.

- **neutral**: if f does not depend on the identity of candidates, i.e., if two candidates are exchanged in every preference ordering in p, the outcome will change accordingly.

- **Pareto-optimal**: if candidate c_A is ranked higher than candidate c_B in all preference orderings, then $c_B \notin f(p)$.

- **reinforcing**: If p_1, p_2 are disjoint profiles and $f(p_1) \cap f(p_2) \neq \emptyset$ then $f(p_1) \cap f(p_2) = f(p_1 \cup p_2)$.

- **monotonic**: if whenever a profile p is changed to p' by having one voter lifting the winning candidate, $f(p) = f(p')$.

Theorem 1 (Young (1975)). *Suppose that V is a voting method that requires voters to rank the candidates. Then, V is anonymous, neutral and reinforcing if and only if the method is a scoring rule.*

According to Theorem 1, majority vote and Borda vote as scoring rules are anonymous, neutral and reinforcing.

Note simply averaging the predictive scores does not satisfy some relevant properties for providing such as anonymity. That means KGE models with higher predictive scores for the top ranked entity would dominate the final decision. Therefore, we do not consider averaging as baseline in our paper.

A *social welfare function* f_w is a mapping from the set of all possible profiles $\mathcal{P}$ to a set of all linear

orders on C. We next introduce some properties of f_w.

f_w is:

- **weakly Paretian**: for $c_1, c_2 \in C$, if $c_1 \prec_i c_2$ for all $i \in N$, then $c_1 \prec c_2$.

- **independent of irrelevant alternatives (IIA)**: if for any $c_1, c_2 \in C$, the relative ranking of c_1 and c_2 only depends on the relative rankings of c_1 and c_2 provided by the voters - but not on how the voters rank some third candidate c_3.

- a **dictatorship**: if there exists a voter $i^* \in N$ such that, for all $c_1, c_2 \in C$, $c_1 \prec_{i^*} c_2$ implies $c_1 \prec c_2$.

Theorem 2 (Arrow (1951)). *When there are three or more alternatives, then every f_w that is weakly Paretian and IIA must be a dictatorship.*

Majority vote and Borda vote are both weakly Paretian and non-dictatorship (Brandt et al., 2016), therefore according to Theorem 2, they are not IIA. However, range vote as a cardinal voting method meet the Arrow's conditions and additionally provide "maximum information" (i.e. provide their opinion of the maximum possible number of candidates) (Vasiljev, 2014; Smith, 2000).

B Proof of Proposition 1

Proof. Given a set of testing queries $\mathcal{T} = \{(q_1, e_1), \ldots, (q_n, e_n)\}$, we let $\hat{y} \in \mathbb{R}^n$, $y_i = T_K(M_\theta^*, tr(q_i, e_i))$ be the vector that contains a 1 if the baseline model regards e_i as a valid answer. Similarly, we let $y' \in \mathbb{R}^n$, $y_i = T_K(M_\theta, tr(q_i, e_i))$ be the corresponding vector for a competing model $M_\theta \in S_\epsilon(M_\theta^*)$.

Let $\mathbf{1} \in \mathbb{R}^n$ be a vector consisting only of ones. Then we can express the proportion of testing triples not ranked in top-K as $\frac{1}{n}||\mathbf{1} - \hat{y}||_1$ and $\frac{1}{n}||\mathbf{1} - y'||_1$ for the baseline and competing model, respectively. We let $\delta(M_A, M_B)$ denote the discrepancy between two models $M_A, M_B \in \mathcal{M}$.

$$\delta(M_A, M_B) :=$$
$$\frac{1}{n}\sum_{\tau \in \mathcal{T}}^{n} \mathbb{1}[T_K(M_A, \tau) \neq [T_K(M_B, \tau)]$$

We can then rewrite

$$\delta(M_\theta^*, M_\theta) = \frac{1}{n}||y' - \hat{y}||_1$$
$$\leq \frac{1}{n}||\mathbf{1} - y'||_1 + \frac{1}{n}||\mathbf{1} - \hat{y}||_1$$
$$= (1 - H_K(M_\theta)) + (1 - H_K(M_\theta^*))$$
$$\leq 2 - H_K(M_\theta^*) + \epsilon - H_K(M_\theta^*),$$

where we used the triangle inequality and symmetry of the L1-norm for the first inequality and the definition of $S_\epsilon(M_\theta^*)$ for the second. Since $\delta_\epsilon(M_\theta^*) = \max_{M_\theta' \in S_\epsilon(M_\theta^*)} \delta(M_\theta^*, M_\theta')$, we have

$$\delta_\epsilon(M_\theta^*) \leq 2 \cdot (1 - H_K(M_\theta^*)) + \epsilon.$$

$\square$

C More Experiment Settings

C.1 Personal Identification Issue in FB15k and FB15k237

While FB15k and FB15k237 contain information about individuals, it typically focuses on well-known public figures such as celebrities, politicians, and historical figures. Since this information is already widely available online and in various public sources, its inclusion in Freebase doesn't significantly compromise individual privacy compared to datasets containing sensitive personal information.

C.2 Change of $S_\epsilon(M_\theta)'$ after Applying Voting Methods

Theoretically, we need to ensure that the aggregated models within the evaluation set S should also have exactly the same ϵ with the original set of competing models $S_\epsilon(M_\theta^*)'$. In order to do that, the pseudocode of evaluating predictive multiplicity should look like following:

Recall from Algorithm 2, we denote $Aggregate(A, S_{agg})$ as a procedure to aggregate rankings predicted by models in S_{agg} using a voting method A (detailed in section 5.1). The result of $Aggregate(A, S_{agg})$ can be viewed as a new KGE model M_{agg} that predicts the aggregated rankings. train(conf(M), $seed$) denotes the training process of a KGE model, which adopts the same training configurations (including the training graph, hyperparameters, etc.) of a pre-trained model M with a specific random seed.

The procedure described in Algorithm 2 can not guarantee to have same ϵ for both S and $S_\epsilon(M_\theta^*)'$,

Algorithm 3 Pseudocode for evaluation (in theory).

Require: $S_\epsilon(M_\theta^*)'$
1:   $S \leftarrow$ An empty set. ▷ Initialize evaluation set
2:   **if** not apply voting methods **then**
3:       $S \leftarrow S_\epsilon(M_\theta^*)'$
4:   **else**
5:       ▷ Aggregation for the baseline model
6:       $S_{agg}^* \leftarrow$ An empty set.
7:       **for** $i \leftarrow 1$ to 10 **do**
8:          $seed_i \leftarrow$ RandomSeed()
9:          $\hat{M}_\theta^* \leftarrow$ train(conf(M_θ^*), $seed_i$).
10:         $S_{agg}^* \leftarrow S_{agg}^* \cup \{\hat{M}_\theta^*\}$
11:       **end for**
12:       $M_{agg}^* \leftarrow$ Aggregate(A, S_{agg}^*).
13:       $S \leftarrow S \cup \{M_{agg}^*\}$.
14:
15:       ▷ Aggregation for the competing models
16:       **while** $|S| \leq 10$ **do**
17:          $S_{agg} \leftarrow$ An empty set.
18:          **do**
19:             **for** $i \leftarrow 1$ to 10 **do**
20:                $seed_i \leftarrow$ RandomSeed()
21:                $\hat{M}_\theta \leftarrow$ train(conf(M_θ^*), $seed_i$).
22:                $S_{agg} \leftarrow S_{agg} \cup \{\hat{M}_\theta\}$
23:             **end for**
24:             $M_{agg} \leftarrow$ Aggregate(A, S_{agg}).
25:          **while** $D(M_{agg}^*, M_{agg}) \leq \epsilon$
26:          $S \leftarrow S \cup \{M_{agg}\}$.
27:       **end while**
28:   **end if**
29:
30:   Evaluate $Hits@K$ for all models in S and report the average value.
31:   Evaluate $\hat{\alpha}_\epsilon$ and $\hat{\delta}_\epsilon$ for S.

since $Hits@K$ changes after applying voting methods. However, obtaining a desirable aggregated model with the do-while loop (from line 19 to line 27) in Algorithm 3 can be very time/computational consuming (approximately 10 hours for each loop). Therefore, we obtain the aggregated model from each competing model in $S_\epsilon(M_\theta^*)'$ to reduce the training effort in Algorithm 2. Empirically, we observe a negligible deviation of ϵ after applying the evaluation procedure of Algorithm 2, see Figure 5. This level of ϵ deviation should not significantly change our claims.
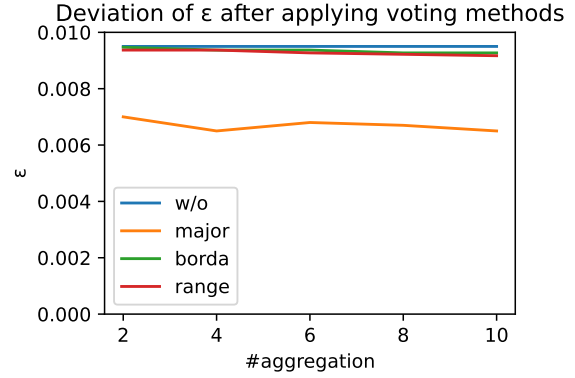

Figure 5: Deviation of ϵ after voting methods wrt. the number of models used for aggregation (results for RESCAL on FB15k237).

C.3   Hyperparameter Search

To get the baseline model M_θ^*, we use PyTorch-based library LibKGE (Broscheit et al., 2020) (MIT-license) and basically follow the hyperparameter search strategy in Ruffinelli et al. (2019). We recall the important details again in this section.

We first conduct quasi-random hyperparameter search via a Sobol sequence, which aims to distribute hyperparameter settings evenly to avoid "clumping" effects (Bergstra and Bengio, 2012). More specifically, for each dataset and model, we generated 30 different configurations per valid combination of training type and loss function. we added a short Bayesian optimization phase (best configuration so far + 30 new trials) to tune the hyperparameters further. All above steps are conducted using Ax framework (https://ax.dev/)

We use a large hyperparameter space including loss functions (pairwise margin ranking with hinge loss, binary cross entropy, cross entropy), regularization techniques (none/L1/L2/L3, dropout), optimizers (Adam, Adagrad), and initialization methods used in the KGE community as hyperparameters. We consider 128, 256, 512 as possible embedding sizes. More details see in Ruffinelli et al. (2019, Table 5).

The hyperparameters of the baseline models are located within the software folder we submitted. Concretely, all configuration files (*.yaml) that we use for training baseline models/competing models/models for aggregation can be found in folder "configs".

C.4   GPU Hours

We use a Linux machine with a 40GB NVIDIA A100 SXM4 GPU. For each KGE methods on one

benchmark dataset, we allocate at most 80 hours to fit the baseline models, 14 hours to construct competing models and 10 hours to fit the models used for aggregation.

D More Experiment Results

Due to the page limit, we represent more experiment results in this section.

D.1 Experiments for Link Prediction in Context of Recommendation

Table 7 presents accuracy and predictive multiplicity metrics for six KGE models across four datasets, extending the findings from Table 5. Key observations are discussed in Section 6.1. Notably, datasets with data leakage, such as WN18 and FB15k, consistently exhibit larger predictive multiplicity metrics compared to datasets without this issue, namely WN18RR and FB15k237. This trend is visualized in Figure 4 and elaborated upon in Section 7.

D.2 Experiments for Link Prediction in Context of Query Answering

We define link prediction as binary classification problem in the main body of the paper, it is suitable for recommendation systems, where people only care about the top-K results. But there are cases where people care more about the answer set of the query. For example, CQD (Arakelyan et al., 2021) decomposite logical queries into one-step atomic queries like $\langle h, r, ?\rangle$ or $\langle ?, r, t\rangle$ and predict the answer set for each atomic query with ComplEx. In this case, We can define link prediction as predicting an answer set A for queries. We denote $tr(q, e)$ as the corresponding triple $\langle h, r, e\rangle$ or $\langle e, r, t\rangle$, respectively.

Definition 9 (Link Prediction for Query Answering). *Given a KGE model M_θ, a query $q \in Q$ and a scoring-based threshold τ, the answer set A of the query q include all entities that have predictive scores exceeding the threshold.*

$$A_\tau(M_\theta, q) = \{e \in E \mid M_\theta(tr(q, e)) \geq \tau\}. \quad (13)$$

Then we adapt all definition of predictive multiplicity and its metrics to this setting. The definition of the ϵ-level set remains the same. Embedding-based query answering exhibits predictive multiplicity if competing models suggest different answer sets for a given query.

Definition 10 (Predictive Multiplicity). *Given a threshold τ, a baseline model M_θ^*, and an error*

tolerance ϵ, the prediction of query q exhibits predictive multiplicity if there exists a model $M_\theta \in S_\epsilon(M_\theta^*)$ such that $A_\tau(M_\theta, q) \neq A_\tau(M_\theta^*, q)$.

Definition 11 (Ambiguity). *Given a testing query set Q' and a threshold τ, the ambiguity of link prediction over the ϵ-level set $S_\epsilon(M_\theta^*)$ is the proportion of testing queries that are provided different answer sets by a competing model $M_\theta \in S_\epsilon(M_\theta^*)$:*

$$\alpha(M_\theta^*) := \frac{1}{|Q'|} \sum_{q \in Q'} \max_{M_\theta \in \mathcal{M}} \mathbb{1}[A_\tau(M_\theta, q) \neq A_\tau(M_\theta^*, q)]. \quad (14)$$

Definition 12 (Discrepancy). *Given a testing query set Q' and a threshold τ, the discrepancy of link prediction over the ϵ-level set $S_\epsilon(M_\theta^*)$ is the maximum proportion of testing queries that are provided different answer sets by a competing model $M_\theta \in S_\epsilon(M_\theta^*)$:*

$$\delta(M_\theta^*) := \max_{M_\theta \in \mathcal{M}} \frac{1}{|Q'|} \sum_{q \in Q'} \mathbb{1}[A_\tau(M_\theta, q) \neq A_\tau(M_\theta^*, q)]. \quad (15)$$

Additionally, we introduce a new evaluation metric *agreement* to measure the overlap of the predicted answer sets from competing models based on Jaccard similarity (Jaccard, 1901). The Jaccard similarity (Jaccard, 1901) between two sets, denoted as $Sim(A, B)$, is defined as the ratio of the cardinality of their intersection to the cardinality of their union.

$$Sim(A, B) := \frac{|A \cap B|}{|A \cup B|} \quad (16)$$

Agreement is then defined as

Definition 13 (Agreement). *Given a testing query set Q' and a threshold τ, the agreement of link prediction over the ϵ-level set $S_\epsilon(M_\theta^*)$ is average Jaccard similarity of predicted answer sets provided by competing models $M_\theta \in S_\epsilon(M_\theta^*)$.*

$$J(M_\theta^*) = \frac{\sum_{q \in Q'} \sum_{M_\theta \in S_\epsilon(M_\theta^*)} Sim(P_\tau(M_\theta, q), P_\tau(M_\theta^*, q))}{|Q'| \cdot |S_\epsilon(M_\theta^*)|}$$

We summarize the results of multiplicity in this setting in Table 8 and 9. We observe more significant predictive multiplicity behavior, since it is more challenging to predict the same answer set from competing models. It requires very robust rankings from competing models. And it heavily relies on the scoring-based threshold. Nevertheless,

Model	Dataset	Baselines	$Hits@10\uparrow$	$\alpha_\epsilon\downarrow$	$\delta_\epsilon\downarrow$	Dataset	Baselines	$Hits@10\uparrow$	$\alpha_\epsilon\downarrow$	$\delta_\epsilon\downarrow$
TransE	WN18	w/o	0.903	0.074	0.029	FB15k	w/o	0.755	0.140	0.053
		major	0.296	0.109	0.051		major	0.202	0.150	0.070
		Borda	0.876	0.028	0.011		Borda	0.751	0.036	0.014
		range	0.907	0.017	0.009		range	0.760	0.032	0.014
	WN18RR	w/o	0.518	0.076	0.034	FB15k237	w/o	0.455	0.385	0.145
		major	0.055	0.096	0.045		major	0.155	0.171	0.081
		Borda	0.482	0.032	0.016		Borda	0.456	0.110	0.044
		range	0.519	**0.017**	**0.009**		range	0.470	0.101	0.041
RotatE	WN18	w/o	0.951	0.026	0.009	FB15k	w/o	0.790	0.086	0.032
		major	0.880	0.031	0.016		major	0.464	0.088	0.044
		Borda	**0.957**	**0.008**	**0.004**		Borda	0.797	0.018	0.008
		range	**0.957**	**0.008**	**0.004**		range	**0.798**	**0.016**	**0.007**
	WN18RR	w/o	0.547	0.195	0.074	FB15k237	w/o	0.520	0.163	0.064
		major	0.413	0.064	0.029		major	0.204	0.104	0.053
		Borda	0.564	0.062	0.028		Borda	0.523	0.039	0.017
		range	**0.578**	0.051	0.022		range	**0.524**	**0.037**	**0.016**
RESCAL	WN18	w/o	0.940	0.039	0.016	FB15k	w/o	0.714	0.217	0.081
		major	0.462	0.015	0.007		major	0.137	0.050	0.024
		Borda	0.935	0.011	0.005		Borda	0.716	0.054	0.022
		range	0.944	0.012	0.005		range	0.729	0.048	0.020
	WN18RR	w/o	0.517	0.248	0.095	FB15k237	w/o	0.482	0.375	0.140
		major	0.198	0.108	0.054		major	0.145	0.165	0.089
		Borda	0.561	0.099	0.043		Borda	0.485	0.107	0.048
		range	0.575	0.084	0.034		range	0.498	0.098	0.042
DistMult	WN18	w/o	0.938	0.044	0.018	FB15k	w/o	0.773	0.170	0.064
		major	0.459	0.012	0.008		major	0.157	0.049	0.023
		Borda	0.927	0.013	0.006		Borda	0.766	0.052	0.021
		range	0.941	0.015	0.007		range	0.778	0.048	0.019
	WN18RR	w/o	0.526	0.169	0.068	FB15k237	w/o	0.476	0.320	0.120
		major	0.185	0.078	0.037		major	0.144	0.124	0.059
		Borda	0.524	0.055	0.024		Borda	0.475	0.088	0.037
		range	0.542	0.048	0.021		range	0.488	0.082	0.034
ComplEx	WN18	w/o	0.941	0.042	0.018	FB15k	w/o	0.765	0.210	0.081
		major	0.458	0.009	0.005		major	0.158	0.047	0.023
		Borda	0.943	0.021	0.010		Borda	0.765	0.076	0.032
		range	0.945	0.020	0.009		range	0.780	0.071	0.029
	WN18RR	w/o	0.541	0.217	0.085	FB15k237	w/o	0.482	0.308	0.116
		major	0.243	0.243	0.126		major	0.145	0.121	0.055
		Borda	0.559	0.067	0.030		Borda	0.480	0.087	0.036
		range	0.573	0.058	0.024		range	0.493	0.082	0.032
ConvE	WN18	w/o	0.938	0.043	0.019	FB15k	w/o	0.766	0.177	0.066
		major	0.476	0.039	0.021		major	0.175	0.085	0.041
		Borda	0.933	0.014	0.005		Borda	0.761	0.052	0.022
		range	0.942	0.015	0.006		range	0.771	0.049	0.020
	WN18RR	w/o	0.500	0.222	0.088	FB15k237	w/o	0.474	0.340	0.130
		major	0.185	0.092	0.047		major	0.150	0.154	0.074
		Borda	0.522	0.082	0.035		Borda	0.474	0.092	0.039
		range	0.534	0.068	0.027		range	0.486	0.085	0.034

Table 7: This table presents the metrics for accuracy (i.e. Hits@K and ϵ) and for predictive multiplicity (i.e. α_ϵ and δ_ϵ) for different voting methods applied on different KGE models and four datasets.

model	dataset	baseline	Hits@1↑	α@1↓	δ@1↓	dataset	baseline	Hits@1↑	α@1↓	δ@1↓
TransE	WN18	w/o	0.499	0.459	0.315	FB15k	w/o	0.659	0.376	0.273
		major	0.494	0.210	0.145		major	0.654	0.180	0.118
		Borda	0.494	0.203	0.135		Borda	0.655	0.159	0.115
		range	0.497	0.194	0.129		range	0.661	0.144	0.100
	WN18RR	w/o	0.105	0.647	0.472	FB15k237	w/o	0.544	0.312	0.206
		major	0.106	0.318	0.218		major	0.543	0.101	0.069
		Borda	0.109	0.308	0.228		Borda	0.544	0.090	0.060
		range	0.112	0.283	0.215		range	0.548	0.091	0.061
RotatE	WN18	w/o	0.880	0.413	0.343	FB15k	w/o	0.664	0.473	0.376
		major	0.872	0.300	0.219		major	0.674	0.271	0.198
		Borda	0.873	0.296	0.229		Borda	0.681	0.263	0.215
		range	0.875	0.285	0.218		range	0.682	0.248	0.188
	WN18RR	w/o	0.219	0.415	0.297	FB15k237	w/o	0.539	0.335	0.221
		major	0.226	0.172	0.115		major	0.538	0.095	0.063
		Borda	0.224	0.187	0.130		Borda	0.540	0.089	0.057
		range	0.231	0.145	0.100		range	0.543	0.096	0.066
RESCAL	WN18	w/o	0.785	0.647	0.483	FB15k	w/o	0.537	0.605	0.496
		major	0.862	0.344	0.250		major	0.584	0.402	0.282
		Borda	0.867	0.340	0.274		Borda	0.595	0.385	0.309
		range	0.868	0.331	0.263		range	0.605	0.374	0.301
	WN18RR	w/o	0.194	0.734	0.639	FB15k237	w/o	0.492	0.635	0.518
		major	0.213	0.510	0.372		major	0.534	0.352	0.249
		Borda	0.232	0.425	0.332		Borda	0.550	0.326	0.252
		range	0.241	0.395	0.318		range	0.556	0.308	0.231
DistMult	WN18	w/o	0.861	0.385	0.325	FB15k	w/o	0.696	0.425	0.349
		major	0.860	0.298	0.208		major	0.695	0.306	0.219
		Borda	0.862	0.310	0.247		Borda	0.697	0.302	0.247
		range	0.862	0.304	0.238		range	0.700	0.298	0.240
	WN18RR	w/o	0.133	0.780	0.675	FB15k237	w/o	0.417	0.817	0.643
		major	0.163	0.517	0.366		major	0.511	0.392	0.271
		Borda	0.171	0.449	0.342		Borda	0.523	0.399	0.297
		range	0.183	0.412	0.293		range	0.543	0.330	0.241
ComplEx	WN18	w/o	0.866	0.379	0.325	FB15k	w/o	0.685	0.420	0.350
		major	0.866	0.310	0.222		major	0.694	0.272	0.206
		Borda	0.866	0.305	0.238		Borda	0.698	0.274	0.218
		range	0.867	0.308	0.238		range	0.700	0.273	0.217
	WN18RR	w/o	0.100	0.957	0.876	FB15k237	w/o	0.419	0.820	0.650
		major	0.158	0.702	0.487		major	0.509	0.393	0.276
		Borda	0.194	0.553	0.438		Borda	0.528	0.398	0.297
		range	0.203	0.489	0.372		range	0.543	0.332	0.256
ConvE	WN18	w/o	0.870	0.435	0.354	FB15k	w/o	0.634	0.568	0.436
		major	0.862	0.302	0.214		major	0.672	0.315	0.219
		Borda	0.863	0.309	0.251		Borda	0.686	0.307	0.240
		range	0.864	0.299	0.239		range	0.688	0.293	0.225
	WN18RR	w/o	0.150	0.617	0.469	FB15k237	w/o	0.520	0.554	0.434
		major	0.164	0.332	0.232		major	0.538	0.283	0.194
		Borda	0.164	0.324	0.240		Borda	0.548	0.255	0.178
		range	0.168	0.288	0.201		range	0.553	0.218	0.145

Table 8: predictive multiplicity evaluation for top-1 answers in query answering setting.

Figure 6: Accuracy for ComplEx on Nations dataset with respect to ϵ.

voting method reduce the number of conflicting prediction also in that settings. In the future work, it is interesting to find out a way to set the threshold properly or at least quantify the uncertainty of the answer set for the threshold.

D.3 Accuracy for ComplEx on Nations dataset with respect to ϵ

See figure 6.

D.4 Complete Results of Investigating the Number of Aggregated Models

Figure 7 - 10 show the results of investigating the predictive multiplicity wrt. the number of aggregated models for all models across all datasets. Figure 11 - 14 show the results of investigating the accuracy wrt. the number of aggregated models for all models across all datasets.

E Further Discussion

Model or predictive multiplicity is not always problematic in some Knowledge Graph Embedding (KGE) applications, as it allows flexibility i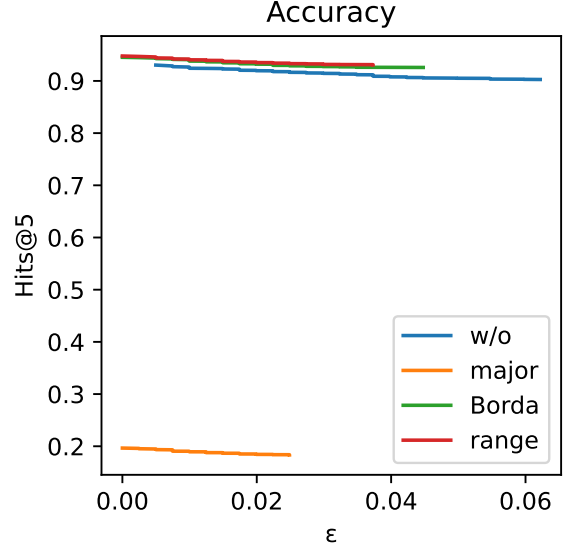n model selection. For instance, KGE methods have been used to approximate statistical reasoning on statistical $\mathcal{EL}$ ($\mathcal{SEL}$) ontologies (Zhu et al., 2023, 2024b; Xiong et al., 2022a). In these cases, the probability intervals of axioms are estimated by assessing the point estimates from multiple KGE models trained with different random seeds. However, if the KGE models are insufficiently diverse due to random sampling from the hypothesis space, these probability intervals may be too narrow. Introducing predictive multiplicity metrics, such as ambiguity and discrepancy, to measure model diversity could improve the sampling process, potentially leading to better approximations for statistical reasoning.

Additionally, previous work has explored logical explanations for KGE predictions. For example, Qu and Tang (2019); Cheng et al. (2022) provide logic rules as explanations, while He et al. (2023, 2024) propose a query embedding model that explains knowledge in the form of $\mathcal{SROI}^-$ description logic axioms. However, due to model multiplicity, these explanations can vary depending on the random seeds used during training. Addressing this variability by finding a set of conflict-free explanations, potentially through the use of computational argumentation frameworks (Potyka, 2021, 2018), is another promising direction for future research. Future work can also be extend to handle knowledge graphs with high-order relational structure (Xiong et al., 2024, 2023; Ding et al., 2024b,a).

model	dataset	baseline	Hits@10↑	α@10↓	δ@10↓	J@10↑	dataset	baseline	Hits@10↑	α@10↓	δ@10↓	J@10↑
TransE	WN18	w/o	0.662	0.940	0.825	0.727	FB15k	w/o	0.468	0.959	0.891	0.597
		major	0.088	0.480	0.376	0.916		major	0.133	0.545	0.469	0.878
		Borda	0.522	0.673	0.505	0.871		Borda	0.463	0.702	0.570	0.850
		range	0.522	0.649	0.497	0.876		range	0.464	0.683	0.549	0.859
	WN18RR	w/o	0.517	0.990	0.930	0.650	FB15k237	w/o	0.239	0.991	0.952	0.564
		major	0.106	0.226	0.177	0.961		major	0.072	0.536	0.447	0.906
		Borda	0.659	0.560	0.428	0.910		Borda	0.242	0.770	0.633	0.865
		range	0.660	0.529	0.401	0.916		range	0.242	0.751	0.611	0.873
RotatE	WN18	w/o	0.730	0.986	0.978	0.391	FB15k	w/o	0.435	0.967	0.934	0.441
		major	0.441	0.355	0.298	0.917		major	0.114	0.654	0.590	0.844
		Borda	0.717	0.902	0.855	0.719		Borda	0.435	0.802	0.712	0.785
		range	0.717	0.868	0.783	0.747		range	0.435	0.790	0.693	0.796
	WN18RR	w/o	0.540	0.976	0.935	0.589	FB15k237	w/o	0.244	0.955	0.870	0.695
		major	0.188	0.270	0.209	0.955		major	0.058	0.424	0.331	0.944
		Borda	0.541	0.765	0.625	0.856		Borda	0.244	0.610	0.457	0.916
		range	0.541	0.723	0.589	0.877		range	0.244	0.586	0.431	0.922
RESCAL	WN18	w/o	0.671	1.000	1.000	0.156	FB15k	w/o	0.345	0.999	0.989	0.334
		major	0.515	0.731	0.623	0.848		major	0.104	0.776	0.714	0.753
		Borda	0.713	0.978	0.951	0.604		Borda	0.386	0.913	0.842	0.671
		range	0.708	0.964	0.918	0.647		range	0.389	0.905	0.828	0.679
	WN18RR	w/o	0.529	0.996	0.995	0.247	FB15k237	w/o	0.210	1.000	0.999	0.236
		major	0.165	0.440	0.400	0.862		major	0.128	0.909	0.873	0.693
		Borda	0.549	0.926	0.852	0.689		Borda	0.239	0.955	0.897	0.672
		range	0.550	0.898	0.817	0.722		range	0.241	0.947	0.881	0.689
DistMult	WN18	w/o	0.701	0.982	0.972	0.343	FB15k	w/o	0.439	0.970	0.936	0.459
		major	0.209	0.271	0.235	0.929		major	0.103	0.697	0.633	0.784
		Borda	0.702	0.922	0.872	0.694		Borda	0.427	0.820	0.725	0.717
		range	0.702	0.883	0.802	0.739		range	0.428	0.805	0.703	0.727
	WN18RR	w/o	0.512	1.000	1.000	0.181	FB15k237	w/o	0.198	1.000	1.000	0.183
		major	0.404	0.534	0.467	0.850		major	0.159	0.974	0.942	0.667
		Borda	0.541	0.973	0.934	0.659		Borda	0.243	0.987	0.958	0.593
		range	0.543	0.958	0.892	0.699		range	0.248	0.974	0.935	0.634
ComplEx	WN18	w/o	0.716	0.985	0.973	0.409	FB15k	w/o	0.420	0.957	0.925	0.427
		major	0.196	0.248	0.220	0.928		major	0.061	0.635	0.582	0.801
		Borda	0.705	0.876	0.784	0.760		Borda	0.423	0.841	0.768	0.711
		range	0.705	0.830	0.731	0.785		range	0.425	0.832	0.753	0.724
	WN18RR	w/o	0.456	1.000	1.000	0.103	FB15k237	w/o	0.197	1.000	1.000	0.172
		major	0.437	0.911	0.839	0.720		major	0.163	0.972	0.946	0.660
		Borda	0.545	0.990	0.966	0.587		Borda	0.246	0.990	0.961	0.588
		range	0.549	0.979	0.941	0.628		range	0.250	0.979	0.950	0.626
ConvE	WN18	w/o	0.713	0.993	0.989	0.277	FB15k	w/o	0.429	0.998	0.990	0.354
		major	0.392	0.379	0.314	0.915		major	0.196	0.800	0.721	0.780
		Borda	0.704	0.939	0.877	0.708		Borda	0.439	0.910	0.833	0.709
		range	0.705	0.912	0.825	0.745		range	0.440	0.892	0.806	0.732
	WN18RR	w/o	0.527	0.995	0.989	0.351	FB15k237	w/o	0.236	0.999	0.989	0.370
		major	0.152	0.393	0.335	0.913		major	0.108	0.815	0.747	0.784
		Borda	0.537	0.905	0.813	0.739		Borda	0.249	0.909	0.813	0.761
		range	0.535	0.873	0.770	0.775		range	0.251	0.893	0.788	0.781

Table 9: predictive multiplicity evaluation for top-10 answers in query answering setting.

E.1 Relationship between Predictive Multiplicity and Entity/Relation Frequency

Figure 15 - 16 demonstrate the relationship between relation frequency and empirical ambiguity/discrepancy.

F AI Assistants In Writing

We use ChatGPT (OpenAI, 2024) to enhance our writing skills, abstaining from its use in research and coding endeavors.
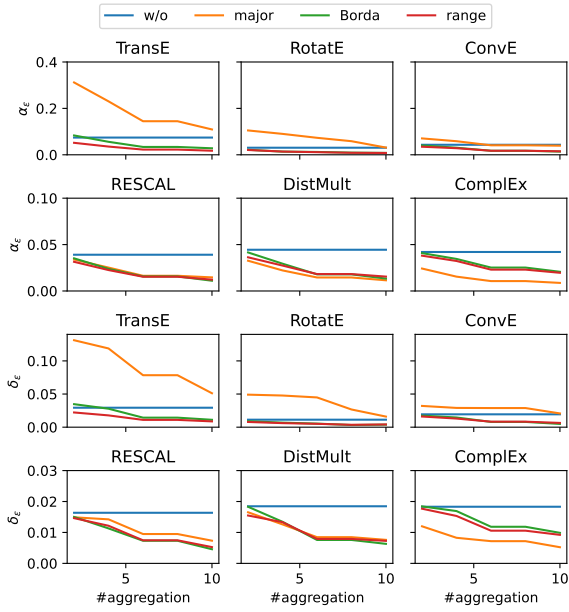
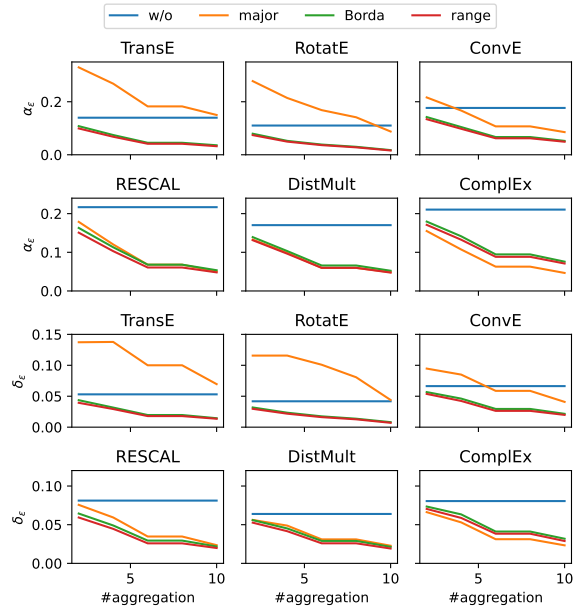Figure 7: Investigation on WN18.

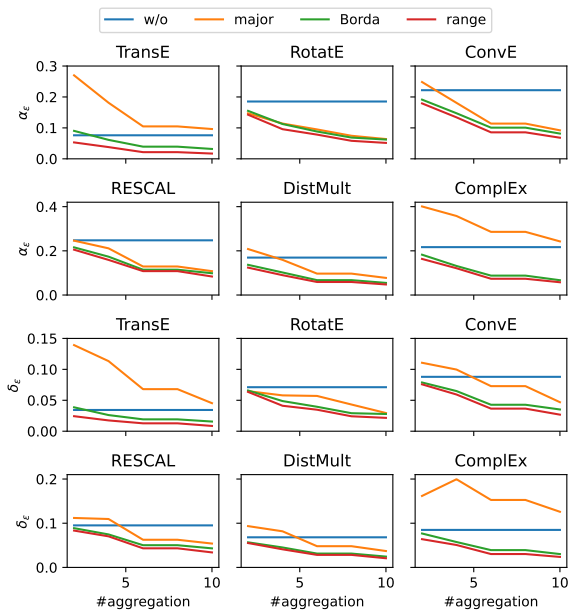Figure 9: Investigation on FB15k.
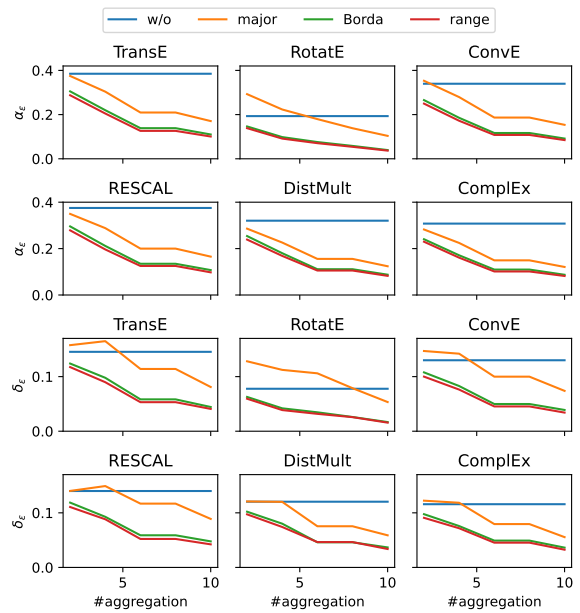

Figure 8: Investigation on WN18RR.
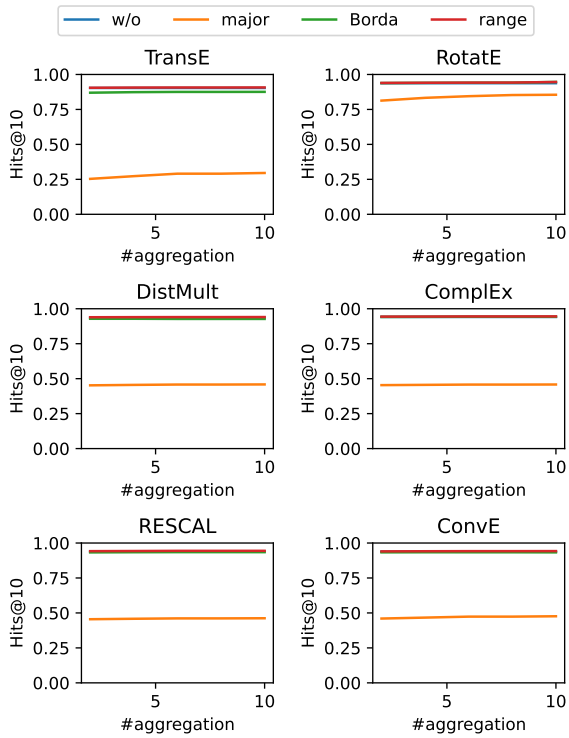

Figure 10: Investigation on FB15k237.

Figure 11: Accuracy investigation on WN18. Note the blue lines (w/o) might be covered by other lines and not visible in diagram.
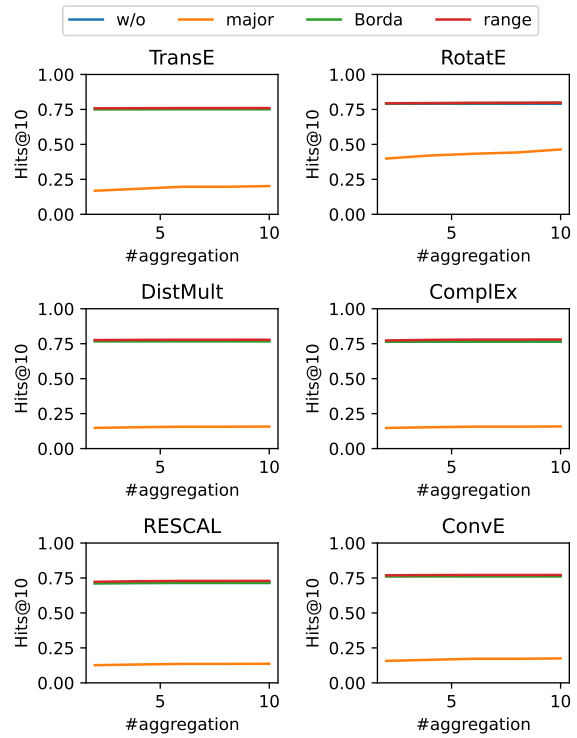

Figure 13: Accuracy investigation on FB15k. Note the blue lines (w/o) might be covered by other lines and not visible in diagram.
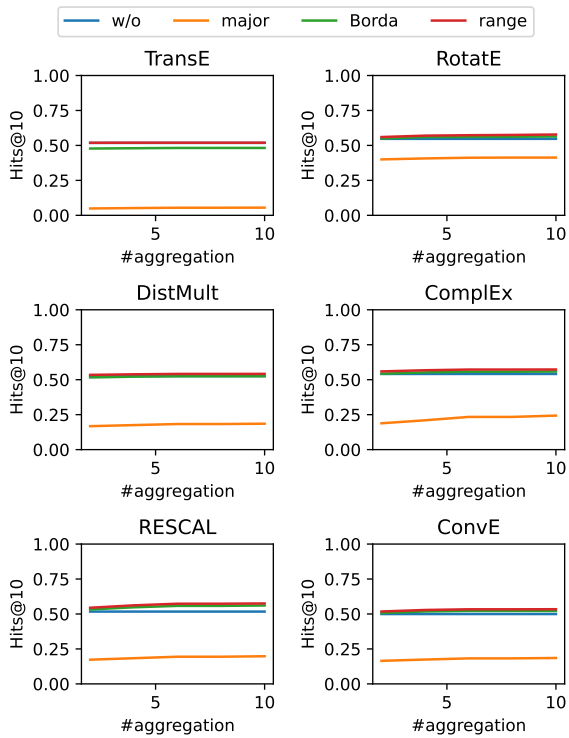

Figure 12: Accuracy investigation on WN18RR. Note the blue lines (w/o) might be covered by other lines and not visible in diagram.
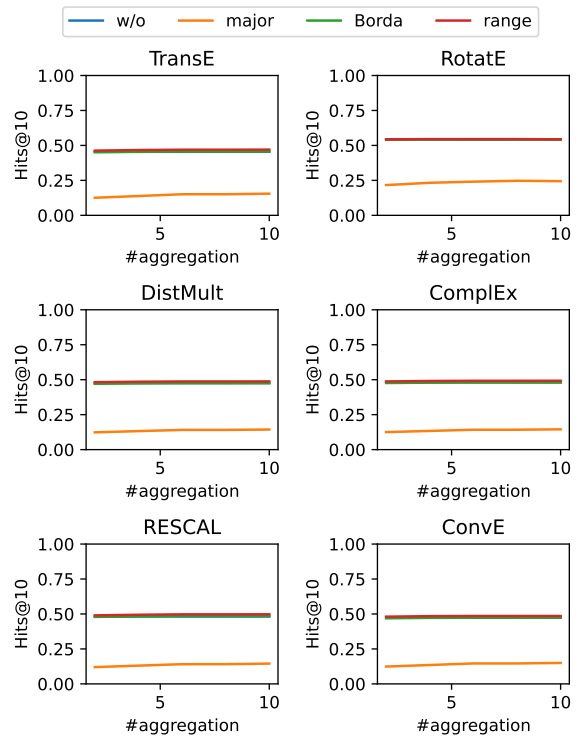

Figure 14: Accuracy investigation on FB15k237. Note the blue lines (w/o) might be covered by other lines and not visible in diagram.
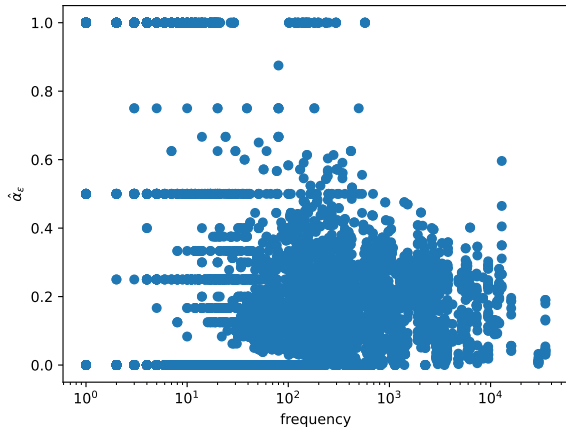
Figure 15: This figure demonstrates the weak negative correlation between relation frequency and empirical ambiguity.
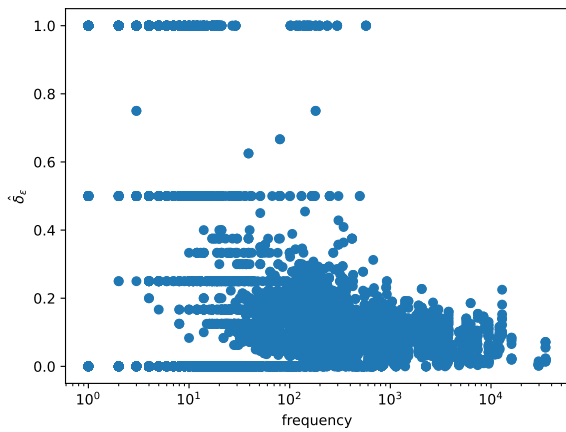

Figure 16: This figure demonstrates the weak negative correlation between relation frequency and empirical discrepancy.