KGE Calibrator+: An Efficient Probability Calibration Method of Knowledge Graph Embedding Models for Trustworthy Link Prediction

Anonymous ACL submission

Abstract

Knowledge graph embedding (KGE) models are designed for the task of link prediction, which aims to infer missing triples by learning accurate representations for entities and relations within a knowledge graph. However, existing KGE research largely overlooks the issue of probability calibration, leading to uncalibrated probability estimates that fail to reflect the true correctness of predicted triples, potentially resulting in erroneous decisions. Moreover, existing calibration methods are not well-suited for KGE models, and no dedicated probability calibration method has been specifically designed for them. In this paper, we propose KGE Calibrator+, the first probability calibration method tailored for KGE models to enhance the trustworthiness of their predictions. To achieve this, we introduce Jump Selection Strategy, which selects the most informative data while filtering out less significant data, and Multi-Binning Scaling, which models different probability levels separately to enhance model capacity and flexibility. Furthermore, we propose a Wasserstein distance-based loss function, improving both calibration performance and optimization stability. Extensive experiments across multiple data sets demonstrate that KGE Calibrator+ consistently outperforms existing calibration methods in terms of both effectiveness and efficiency, making it a promising solution for probability calibration in KGE models.

1 Introduction

Knowledge graphs (KGs) are essential resources for a wide range of knowledge-driven tasks, including semantic search (Xiong et al., 2017), knowledge reasoning (Liu et al., 2021), question answering (Shen et al., 2019; Ye et al., 2023), and reading comprehension (Yang et al., 2019; Meng et al., 2023). Prominent large-scale KGs such as YAGO (Suchanek et al., 2007), DBpedia (Lehmann et al., 2015), and Freebase (Bollacker et al., 2008) encompass millions of entities and hundreds of millions of relational facts, which are typically structured as sets of <*head entity, relation, tail entity*> triples.

However, constructing KGs often involves challenges such as extraction errors and limited input resources, leading to incomplete KGs. This limitation underscores the significance of the link prediction task, also known as knowledge graph completion. By predicting missing links and uncovering new facts, this task plays a pivotal role in addressing the inherent incompleteness of KGs, thereby enhancing their overall quality and practical utility.

Knowledge graph embedding (KGE) models, such as TransE (Bordes et al., 2013), ComplEx (Trouillon et al., 2016), and RotatE (Sun et al., 2019), have been developed to address this challenge. These models aim to learn latent representations of entities and relations within a KG to facilitate the prediction of new facts. Typically, KGE models employ different scoring functions to assign a plausibility score to each triple, ensuring that positive triples are assigned higher scores than negative ones. Beyond link prediction, KGE models have demonstrated remarkable success across diverse applications, including entity alignment (Sun et al., 2018), link-based clustering (Gad-Elrab et al., 2020), and canonicalization (Shen et al., 2022).

While the accuracy of KGE models has seen significant advancements over the past decade, the critical issue of probability calibration remains largely overlooked. Specifically, a KGE model should provide a calibrated probability in addition to its prediction. However, existing studies (Pezeshkpour et al., 2020; Tabacof and Costabello, 2020) have demonstrated that many KGE models are uncalibrated. In practice, these models fail to provide reliable probabilities for the predicted triples, instead producing uncalibrated outputs. This limitation arises because link prediction is fundamentally framed as a "learning to rank" problem. Conse-

quently, the primary evaluation metrics, such as Hits@N and Mean Reciprocal Rank (MRR), focus solely on the relative ranking of candidate triples. These metrics reward KGE models for maintaining correct relative orderings but do not penalize them for assigning excessively high absolute scores to negative triples. As a result, KGE models can achieve strong relative ranking performance while producing uncalibrated probabilities for their predictions. This drawback significantly restricts the applicability of KGE models in high-stakes domains, such as drug discovery (Zeng et al., 2022) and protein targets discovery (Mohamed et al., 2020). In these scenarios, users require both accurate predictions and trustworthy probabilities to assess the reliability of the model's outputs.

To address this critical issue, increasing attention has been directed toward the probability calibration task of KGE models, which aims to convert the uncalibrated scores assigned to candidate triples by KGE models into accurate calibrated probabilities. For instance, if a KGE model predicts that 100 candidate triples each have a 90% probability of being true, then, ideally, 90 of those triples should indeed be correct. If this proportion deviates significantly from 90%, then the model is considered uncalibrated (the model is under-confident if the proportion is higher, and over-confident if it is lower). Intuitively, KGE probability calibration functions as a post-processing technique aimed at enhancing the trustworthiness of link prediction results, thereby offering significant potential benefits for downstream applications.

Despite its importance, probability calibration in KGE models remains an unresolved challenge. Existing studies (Tabacof and Costabello, 2020; Pezeshkpour et al., 2020) have shown that widely used KGE models lack proper calibration, leading to untrustworthy probability estimates. While prior research (Safavi et al., 2020; Zhu et al., 2022) has explored various off-the-shelf calibration techniques such as Platt Scaling, Isotonic Regression, Temperature Scaling, and Histogram Binning, these methods were primarily designed for traditional machine learning models and have not been specifically adapted for KGE models. Some studies have examined calibration in specific tasks, such as triple classification (Tabacof and Costabello, 2020), relation prediction (Safavi et al., 2020), and entity prediction under low-dimensional settings (Wang et al., 2021). However, none of the aforementioned works propose a dedicated calibration method specifically designed for KGE models, leaving a critical gap in this area of research.

To fill this gap, we propose KGE Calibrator+ (KGEC+), the first probability calibration method tailored specifically for KGE models. To enhance training efficiency and reduce noise, we introduce the Jump Selection Strategy, which selects the most informative data while discarding less significant data. Based on the temperature scaling, to improve its expressiveness, we propose Multi-Binning Scaling, which models different probability levels separately, thereby increasing model capacity and flexibility. Furthermore, to further enhance performance and accelerate training, we replace the traditional KL divergence with a loss function based on Wasserstein distance, which provides a more stable and effective optimization process. To the best of our knowledge, this is the first time to leverage the Wasserstein distance in calibration.

Contributions. Our major contributions can be summarized as follows:

- We analyze nine widely used post-processing calibration methods and find that four of them are unsuitable for entity link prediction due to their poor performance, which alters the original link prediction results after calibration.
- We propose KGEC+, the first probability calibration method specifically designed for KGE models, addressing their unique challenges in probability calibration.
- A thorough experimental study over four data sets demonstrates that our method outperforms all baseline methods on the link prediction probability calibration task in terms of both performance and efficiency.

2 Related Work

Probability Calibration in KGE Models. Several studies have investigated probability calibration in KGE models, highlighting their lack of well-calibrated probability estimates. (Tabacof and Costabello, 2020) and (Pezeshkpour et al., 2020) demonstrated that popular KGE models are uncalibrated in the triple classification task. To mitigate this issue, (Tabacof and Costabello, 2020) applied Platt Scaling (Platt et al., 1999) and Isotonic Regression (Zadrozny and Elkan, 2002), while (Safavi et al., 2020) explored Matrix Scaling and Vector Scaling (Guo et al., 2017) in the relation prediction task. (Zhu et al., 2022) conducted a broader evaluation of off-the-shelf calibration techniques,

testing Histogram Binning (Zadrozny and Elkan, 2001), Beta Calibration (Kull et al., 2017), and Temperature Scaling (Guo et al., 2017) in triple classification. While these methods show promise, they were not explicitly designed for KGE models and have limited adaptability to their specific requirements.

Calibration in Specific KGE Tasks. Several studies have focused on calibrating KGE models for specific tasks. For instance, (Wang et al., 2021) examined entity prediction under low-dimensional settings and introduced a causal intervention-based plugin to replace the sigmoid function, which was subsequently calibrated using Platt Scaling or Isotonic Regression. Additionally, (Rao, 2021) explored calibration in KGE models under both the closed-world and open-world assumptions.

While these works contribute to understanding calibration in KGE models, they primarily adapt existing methods rather than proposing dedicated calibration solutions tailored to the unique properties of knowledge graph embeddings. As a result, no existing approach directly addresses the probability calibration needs of KGE models, leaving an important research gap.

3 Preliminaries

3.1 Knowledge Graph

A knowledge graph (KG) $\mathcal{G} = \{\xi\}$ contains a set of triples $\xi = (h, r, t)$, where each triple includes a head entity $h \in \mathcal{E}$, a tail entity $t \in \mathcal{E}$, and a relation $r \in \mathcal{R}$ connecting head and tail. $\mathcal{E}$ and $\mathcal{R}$ refer to the set of all entities and relations of $\mathcal{G}$ respectively. $N = |\mathcal{E}|$ and $M = |\mathcal{R}|$ denote the number of entities and relations respectively.

3.2 Knowledge Graph Embeddings

Knowledge graph embedding (KGE) models aim to represent each head entity h, relation r, and tail entity t from a KG $\mathcal{G}$ as d-dimension continuous embeddings $\mathbf{h}$, $\mathbf{r}$, and $\mathbf{t} \in \mathbb{R}^d$. Each KGE model defines a model-specific score function ψ that assigns a score to each triple $\xi = (h, r, t)$ based on its corresponding embeddings, i.e., $\psi(\xi) = \psi(\mathbf{h}, \mathbf{r}, \mathbf{t})$. Table 1 lists the score functions of the most popular models.

3.3 Link Prediction

Link prediction, the most commonly used task for KGE models, comprises two subtasks: entity prediction and relation prediction. Among these, entity prediction is more challenging than relation prediction due to the larger number of candidate entities that need to be scored and ranked. For instance, in the widely used WN18 dataset (Bordes et al., 2013), there are 40,943 entities but only 18 relations In this paper, we focus on the more challenging entity prediction task.

Table 1: Score functions of popular KGE models, where $\|\cdot\|$ denotes the L_1 norm, $\langle\cdot\rangle$ denotes the generalized dot product, $\mathbf{t}^*$ denotes the complex conjugate of $\mathbf{t}$, Re refers to the real part of a complex number, and $\circ$ denotes the Hadamard product.

KGE model	Score function
TransE (Bordes et al., 2013)	$-\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|$
DistMult (Yang et al., 2015)	$\langle\mathbf{r},\mathbf{h},\mathbf{t}\rangle$
ComplEx (Trouillon et al., 2016)	$Re(\langle\mathbf{r},\mathbf{h},\mathbf{t}^*\rangle)$
RotatE (Sun et al., 2019)	$-\|\mathbf{h}\circ\mathbf{r}-\mathbf{t}\|$

To be specific, the entity prediction task consists of head entity prediction and tail entity prediction. In head entity prediction, given a query of the form $(?, r, t)$, each entity $e_i \in \mathcal{E}$ becomes a potential candidate for the head entity. The trained KGE model assigns a score $\psi(\xi_i)$ to each triple $\xi_i = (e_i, r, t)$, where e_i is a candidate head entity, and r and t are the given relation and tail entity. These scores are then ranked, with higher-ranked triples being more plausible, indicating that the corresponding entity e_i is a likely answer to the query $(?, r, t)$. The task of tail entity prediction could be defined in a similar manner.

4 KGE Calibrator+ Method

In this section, we present our proposed KGE Calibrator method. In Section 4.1, we introduce the Jump Selection Strategy, a technique for selecting the most informative data to improve calibration efficiency. In Section 4.2, we describe Multi-Binning Scaling, which enhances probability calibration by modeling different probability levels separately. Finally, in Section 4.3, we introduce our two calibration methods: KGE Calibrator and its improved version, KGE Calibrator+.

4.1 Jump Selection Strategy

To improve training efficiency and reduce noise, it is crucial to focus on the most informative data while discarding less significant data when training the calibration method. Inspired by (Shen et al., 2022), we propose the Jump Selection Strategy, which selects the most significant data for training rather than using all available data. This Jump

Algorithm 1 Jump Selection Strategy

Input: Query triple $(?, r, t)$, candidate entities $\mathcal{E} = \{e_1, ..., e_i, ..., e_N\}$, KGE model ψ

1: **Generate** candidate triples: $\xi_i \leftarrow (e_i, r, t)$, for $i = 1, \ldots, N$
2: **Compute** uncalibrated scores: $x_i \leftarrow \psi(\mathbf{e}_i, \mathbf{r}, \mathbf{t})$, for $i = 1, \ldots, N$
3: Form score vector: $X \leftarrow \{x_1, ..., x_i, ..., x_N\}$
4: **Compute** probabilities: $P \leftarrow \sigma_{\mathrm{SM}}(X)$
5: **Sort** P in descending order to obtain $\tilde{P}$, such that $\tilde{P}_i$ is the ith largest probability
6: **for** $i = 1$ **to** $N - 1$ **do**
7: $\quad J_i \leftarrow D_{KL}(\tilde{P}_i \parallel \tilde{P}_{i+1})$
8: **end for**
9: $J^* \leftarrow \arg\max_i J_i$
10: $p^* \leftarrow \tilde{P}_{J^*}$

Output: Selected index J^* and its corresponding probability p^* for calibration

Selection Strategy is summarized in Algorithm 1, and we elaborate it as follows.

Given a query $(?, r, t)$, the set of candidate entities $\mathcal{E} = \{e_1, ..., e_i, ..., e_N\}$, and a KGE model ψ, we first generate a set of candidate triples $\Xi = \{\xi_1, ..., \xi_i, ..., \xi_N\}$, where $\xi_i = (e_i, r, t)$ (w.r.t. line 1 in Algorithm 1). Next, we compute the uncalibrated scores for these candidate triples using the KGE model's score function: $X = \{x_1, ..., x_i, ..., x_N\}$, where $x_i = \psi(\xi_i) = \psi(\mathbf{e}_i, \mathbf{r}, \mathbf{t})$ (w.r.t. line 2 in Algorithm 1). These uncalibrated scores are then transformed into uncalibrated probabilities via the softmax function σ_{SM} (w.r.t. line 4 in Algorithm 1). We then sort P in descending order, ensuring that higher probabilities appear first (w.r.t. line 5 in Algorithm 1). Subsequently, for each sorted uncalibrated probability, we compute its Jump Measure J_i using the Kullback–Leibler (KL) divergence D_{KL} between consecutive entries (w.r.t. line 7 in Algorithm 1). Finally, we select the index i corresponding to the maximum Jump Measure J_i, along with its associated probability p, which contains the most valuable information for subsequent calibration (w.r.t. line 9 and line 10 in Algorithm 1). Overall,

4.2 Multi-Binning Scaling

Temperature scaling (Guo et al., 2017) is a particularly appealing post-hoc calibration method because it preserves the accuracy of the original model while transforming its uncalibrated probabilities. However, it suffers from the limited expressiveness, as its model capacity is constrained by fitting only a single scalar parameter $T > 0$ to transform all probabilities, regardless of their magnitude (e.g., 0.1 and 0.9). We hypothesize that the inherent limitation of temperature scaling arises from its inability to separately model different probability levels, restricting its effectiveness in calibration.

While non-parametric methods, such as histogram binning (Zadrozny and Elkan, 2001), offer greater flexibility, they often fail to maintain model accuracy, and the calibrated model's accuracy may degrade significantly. To leverage the advantages of histogram binning while mitigating its drawbacks, we divide all uncalibrated probabilities $p^* = \{p_1^*, ..., p_i^*, ..., p_N^*\}$ into mutually exclusive bins $B1, \ldots, B_m, ..., B_M$. Each bin is assigned a unique scalar temperature parameter T_m, such that if p_i^* belongs to bin B_m, its transformation is given by:

$$\hat{p}_i = \sigma_{SM}(p_i^*/T_m^2), \tag{1}$$

where σ_{SM} is the softmax function. To ensure a structured binning process, we define bin boundaries for a suitably chosen M as follows:

$$0 = a_1 \leq a_2 \leq ... \leq a_{M+1} = 1, \tag{2}$$

where the bin B_M corresponds to the interval $(a_m, a_{m+1}]$. For simplicity, the bin boundaries are chosen to be equal length intervals.

For data points that are not selected by the Jump Selection Strategy, we uniformly apply the temperature parameter T_m corresponding to p^*. This ensures that the overall model accuracy remains unaffected while still benefiting from the calibrated probability adjustments. Overall, this approach integrates the benefits of temperature scaling (which preserves accuracy) with the flexibility of histogram binning, allowing for a more expressive and effective probability calibration approach.

4.3 Optimization

4.3.1 KGE Calibrator

After applying the Jump Selection Strategy to identify representative data and using Multi-Binning Scaling to determine multi-scale bins, we train the temperature parameters T_m using the Kullback–Leibler (KL) divergence as loss function:

$$D_{KL}(p^* \parallel q) = \sum_i p_i^* \log \frac{p_i^*}{q_i}, \tag{3}$$

where q is the golden label. We refer to this method as KGE Calibrator (KGEC).

Notably, unlike standard temperature scaling, we do not impose the constraint $T > 0$; instead, we use T_m^2 to ensure stability. This modification not only preserves model accuracy but also enables fine-grained probability calibration across different probability scales. A theoretical justification for this method is provided in Appendix A.1.

4.3.2 KGE Calibrator+

There are two key disadvantages of using KL divergence as the loss function in KGEC: (1) Since the number of candidate entities N in this task is large, the probability p approaches 0, leading to a loss value of 0. This reduces the effective utilization of data and negatively impacts calibration performance. (2) When the label q is 0, the loss becomes infinite, which can easily cause gradient explosion, thereby slowing down convergence. For a detailed analysis, refer to Appendix C.

To address these issues, we propose using the Wasserstein distance instead of KL divergence as the loss function for KGE Calibrator+ (KGEC+). The Wasserstein distance measures the minimum cost required to transform one probability distribution into another by modeling it as an optimal transport (OT) problem. It considers the set of a transportation polytope $U(p^*, q)$, which contains all nonnegative transport matrices P:

$$U(p^*, q) = \{\mathbf{P} \in \mathbb{R}_+^{d\times d} | \mathbf{P}\mathbf{1}_d = p^*, \mathbf{P}^\top \mathbf{1}_d = q\}, \tag{4}$$

where $1_d \in \mathbb{R}^d$ is a vector of ones.

Given a cost matrix $M \in \mathbb{R}^{d\times d}$, the Wasserstein distance is defined as the minimum transport cost required to map p^* to q using the transport matrix P.

$$D_{WD}(p^*, q) = \min_{P\in U(p^*,q)} \langle P, M\rangle = \sum_{m,n} P_{m,n} M_{m,n}, \tag{5}$$

where $\langle \cdot, \cdot \rangle$ stands for the Frobenius dot-product and $M_{m,n} = |p_m^* - q_n|$ represents the absolute difference between the m-th and n-th elements of p^* and q.

To improve computational efficiency, we use the Sinkhorn distance (Cuturi, 2013), which provides a fast approximation to the constrained Wasserstein distance by introducing entropy regularization. Given the OT plan P^λ and cost matrix M, the Sinkhorn distance is defined as follows:

$$D_{SD}(p^*, q) = \left\langle P^\lambda, M \right\rangle, \tag{6}$$

where $\lambda > 0$ is the weight for entropy regularization. The OT plan P^λ is obtained by solving:

$$P^\lambda = \underset{P\in U(p^*,q)}{\arg\min} \langle P, M\rangle - \frac{1}{\lambda} h(P), \tag{7}$$

where $h(P)$ is the entropy of P. The solution P^λ computed iteratively via Sinkhorn normalization (Cuturi, 2013) as follows:

$$\begin{aligned} u^t &= p^* \oslash (K^\top v^{t-1}), \\ v^t &= q \oslash (K u^t), \end{aligned} \tag{8}$$

where $\oslash$ indicates element-wise division, t denotes the iteration time, and $K = \exp(-\frac{M}{\lambda})$ is the kernel matrix with entropy regularization weight λ. Finally, the optimal transport plan P^λ is given by:

$$P^\lambda = \operatorname{diag}(v^t) K \operatorname{diag}(u^t), \tag{9}$$

By leveraging Sinkhorn distance, KGEC+ achieves a more robust and efficient probability calibration process, avoiding the numerical instability issues associated with KL divergence while maintaining computational feasibility.

5 Experiments

For the experiments, we first introduce three key research questions (RQs), and then use our experimental results to address each of these questions individually.

- **RQ1**: Which of the existing post-processing calibration methods can not affect the KGE results?
- **RQ2**: Can our proposed KGE Calibrators surpass the performance of existing methods without changing the KGE results?
- **RQ3**: Are our proposed KGE Calibrators method efficient?

Section 5.1 details the datasets used in our experiments, along with the training and learning processes for both the link prediction models and calibration functions. Section 5.2 presents the accuracy evaluation for **RQ1**. Section 5.3 presents the effectiveness evaluation for **RQ2**. Section 5.4 discusses the time for **RQ3**.

5.1 Experimental Setting

5.1.1 Data sets

We evaluate our proposed model on four popular data sets, which are commonly used to evaluate link prediction, where FB15K (Bordes et al., 2013) and FB15K-237 (Toutanova and Chen, 2015) were extracted from Freebase (Bollacker et al.,

Table 2: Statistics of the used KGE data sets.

Data set	*#Entity*	*#Relation*	*#Training*	*#Validation*	*#Testing*
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

2008), WN18 (Bordes et al., 2013) and WN18RR (Dettmers et al., 2018) were extracted from WordNet (Miller, 1995). Note that FB15K-237 and WN18RR are subsets of FB15K and WN18, respectively, in which near-same and near-reverse relations have been removed. These datasets are publicly available, and already partitioned into training, validation and testing splits. The statistics of them are summarized into Table 2.

5.1.2 KGE models

To evaluate our proposed model, we leverage four famous KGE models in our experiments, i.e., TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), and RotatE (Sun et al., 2019) The score functions of them are shown in Table 1. It is noted that any KGE models could be employed as the input of our model, as long as it could encode triples into embeddings and get their scores. Therefore, choosing different KGE models is not the focus of this paper and left for future exploration.

5.1.3 Calibration baselines

All calibration baselines are listed as follows.

• Platt Scaling (PS) (Platt et al., 1999) is a parametric approach to calibration, which is based on transforming the non-probabilistic outputs of a binary classifier to calibrated confidence scores.

• Histogram Binning (HB) (Zadrozny and Elkan, 2001) is a simple non-parametric calibration method. All uncalibrated predictions are divided into mutually exclusive bins, where each bin is assigned a calibration score.

• Isotonic Regression (IR) (Zadrozny and Elkan, 2002) is a strict generalization of histogram binning in which the bin boundaries and bin predictions are jointly optimized.

• Bayesian Binning into Quantiles (BBQ) (Naeini et al., 2015) is a extension of histogram binning using the concept of Bayesian model averaging.

• Matrix Scaling (MS) and Vector Scaling (VS) (Guo et al., 2017) are two multi-class extensions of Platt scaling.

• Temperature Scaling (TS) (Guo et al., 2017) is the simplest extension of Platt scaling, uses a single scalar parameter $T > 0$ for all candidates.

• Meta-Cal (Ma and Blaschko, 2021) integrates bipartite-ranking model with selective classification to improve calibration map.

• Parametrized Temperature Scaling (PTS) (Tomani et al., 2022) is the generalization of temperature scaling by computing prediction-specific temperatures, parameterized by a neural network.

In order to keep the accuracy of the KGE model as unchanged as possible, we only use post-hoc techniques for KGE models. Therefore, other calibration techniques such as regulization (Ahn et al., 2019), ensemble (Lakshminarayanan et al., 2017), MC-dropout (Gal and Ghahramani, 2016) and mixup (Thulasidasan et al., 2019) are not within the scope of this paper. We fail to obtain the experimental results of Beta Calibration (Kull et al., 2017), since it needs too much time to execute. For example, for the smallest data set, i.e., WN18RR, it needs more than 60 hours.

5.1.4 Evaluation measures

Calibrating a model requires reliable metrics to detect miscalibration, and effective techniques to fix such distortion. To evaluate results from different perspectives, we utilize Expected Calibration Error (ECE) (Naeini et al., 2015), Adaptive Calibration Error (ACE) (Nixon et al., 2019) and Negative Log Likelihood Metric (NLL) as metrics for evaluating the performance of calibration methods. Due to the limited space, we omit the detailed computing methods of these metrics and you could refer to (Naeini et al., 2015; Nixon et al., 2019) for more details. ECE, ACE, and NLL metrics evaluate results from different perspectives. To give an overall evaluation of each method, we calculate the average of each metric for different data set and different KGE models as **Average**, which is a standard comprehensive metric for the task of KGE calibration.

5.1.5 Setting details

To ensure a fair comparison, all baselines and metrics we use are from third-party frameworks or their original codes. Specifically, the code of PS, HB, IR, BBQ, and TS are from net:cal[1]. The code of MS and VS and all metrics are calculated by the TorchUncertainty[2]. The code of Meta-Cal[3] and

[1] https://efs-opensource.github.io/calibration-framework/build/html/index.html

[2] https://torch-uncertainty.github.io

[3] https://github.com/maxc01/metacal/tree/master

PTS[4] is from their official code. For both KGEC and KGEC+, the number of bins is set to 10, the learning rate is set to 0.01, the number of iterations is set to 100, the temperature for each bin is initially set as 1 and the optimizer is AdamW (Loshchilov and Hutter, 2019). Except for VS, MS, and TS which uses the *Multiclass* setting, all other baselines use the *One-vs-all* setting to avoid unacceptable training time. We follow the closed world assumption in our experiments. This is because the open world assumption requires a label for each triplet, which is missing in existing data sets. All experimental results are the average values obtained after running 10 times. We make the source code used in this paper publicly available for future research[5].

5.2 Accuracy Affection Study for RQ1

Table 3 presents the results of the TransE model across various datasets after applying different calibration methods. The Uncal row represents the original, uncalibrated results, ↑ indicates an improvement, while ↓ indicates a decline compared to the original uncalibrated results. Among the reported evaluation metrics: A lower Mean Rank (MR) indicates better performance. Higher values of Mean Reciprocal Rank (MRR), HITS@1, HITS@3, and HITS@10 indicate better performance.

From the experimental results in Table 3, we can see that (1) HB, IR, BBQ, MS, and Meta-Cal significantly degrade performance across all four datasets, making them unsuitable as calibrators for KGE models in the entity link prediction task; (2) KGEC and KGEC+ maintain model accuracy across all datasets, demonstrating their effectiveness as the most suitable calibration methods for this task; (3) PS, VS, and TS either preserve or slightly improve accuracy on WN18 and WN18RR and generally do not lead to performance deterioration; (4) VS slightly degrades performance on FB15K and PTS on WN18, but given that the decline is minor and it performs well on other datasets, its overall impact remains acceptable.

5.3 Effectiveness Study for RQ2

Table 4 presents the impact of different calibration methods on the performance of various KGE models across multiple datasets. Notably, baselines such as HB, IR, BBQ, MS, and Meta-Cal are excluded, as they were shown to degrade the original accuracy in Section 5.2. Since a calibration method that reduces accuracy is impractical, these baselines are omitted from further evaluation.

Overall, Table 4 demonstrates that our proposed KGEC+ method consistently outperforms all competitive baselines in terms of average ECE, ACE and NLL across all four datasets. Here are the key observations from Table 4: (1) Poor performance of simple baselines: The three simple calibration methods (PS, VS, and TS) perform poorly, often yielding worse results than the original, uncalibrated models across all data sets. This is

Table 3: Effect of different calibration methods on the performance of the TransE model across various datasets.

Method	*MR*	*MRR*	*HITS@1*	*HITS@3*	*HITS@10*
			WN18		
Uncal	263	0.772	0.706	0.807	0.920
PS	260 ↑	0.772	0.706	0.807	0.920
HB	15299 ↓	0.225 ↓	0.212 ↓	0.236 ↓	0.240 ↓
IR	14590 ↓	0.251 ↓	0.232 ↓	0.267 ↓	0.279 ↓
BBQ	15178 ↓	0.218 ↓	0.200 ↓	0.233 ↓	0.244 ↓
VS	258 ↑	0.772	0.706	0.807	0.920
MS	16483 ↓	0.013 ↓	0.005 ↓	0.013 ↓	0.029 ↓
TS	260 ↑	0.772	0.706	0.807	0.920
Meta-Cal	1784 ↓	0.718 ↓	0.657 ↓	0.749 ↓	0.856 ↓
PTS	2116 ↓	0.751 ↓	0.706	0.775 ↓	0.849 ↓
KGEC	263	0.772	0.706	0.807	0.920
KGEC+	263	0.772	0.706	0.807	0.920
			WN18RR		
Uncal	3437	0.223	0.014	0.401	0.528
PS	3437	0.223	0.014	0.401	0.528
HB	19455 ↓	0.071 ↓	0.053 ↑	0.087 ↓	0.099 ↓
IR	18143 ↓	0.102 ↓	0.080 ↑	0.119 ↓	0.139 ↓
BBQ	18196 ↓	0.071 ↓	0.050 ↑	0.085 ↓	0.105 ↓
VS	3421 ↑	0.224 ↑	0.014	0.401	0.529 ↑
MS	18178 ↓	0.009 ↓	0.003 ↓	0.008 ↓	0.020 ↓
TS	3437	0.223	0.014	0.401	0.528
Meta-Cal	3437	0.223	0.014	0.401	0.528
PTS	3437	0.223	0.014	0.401	0.528
KGEC	3437	0.223	0.014	0.401	0.528
KGEC+	3437	0.223	0.014	0.401	0.528
			FB15K		
Uncal	40	0.731	0.646	0.793	0.865
PS	40	0.731	0.646	0.793	0.865
HB	2275 ↓	0.570 ↓	0.510 ↓	0.614 ↓	0.670 ↓
IR	982 ↓	0.615 ↓	0.530 ↓	0.675 ↓	0.761 ↓
BBQ	1275 ↓	0.589 ↓	0.509 ↓	0.646 ↓	0.726 ↓
VS	41 ↓	0.730 ↓	0.646	0.791 ↓	0.862 ↓
MS	3687 ↓	0.038 ↓	0.024 ↓	0.039 ↓	0.061 ↓
TS	40	0.731	0.646	0.793	0.865
Meta-Cal	1149 ↓	0.677 ↓	0.604 ↓	0.735 ↓	0.787 ↓
PTS	40	0.731	0.646	0.793	0.865
KGEC	40	0.731	0.646	0.793	0.865
KGEC+	40	0.731	0.646	0.793	0.865
			FB15K-237		
Uncal	173	0.330	0.231	0.368	0.527
PS	173	0.330	0.231	0.368	0.527
HB	3497 ↓	0.289 ↓	0.224 ↓	0.321 ↓	0.416 ↓
IR	2141 ↓	0.309 ↓	0.234 ↑	0.343 ↓	0.455 ↓
BBQ	2335 ↓	0.280 ↓	0.209 ↓	0.310 ↓	0.422 ↓
VS	173	0.330	0.231	0.368	0.527
MS	3704 ↓	0.033 ↓	0.014 ↓	0.032 ↓	0.070 ↓
TS	173	0.330	0.231	0.368	0.527
Meta-Cal	1231 ↓	0.308 ↓	0.218 ↓	0.344 ↓	0.490 ↓
PTS	173	0.330	0.231	0.368	0.527
KGEC	173	0.330	0.231	0.368	0.527
KGEC+	173	0.330	0.231	0.368	0.527

[4] https://github.com/tochris/pts-uncertainty

[5] https://anonymous.4open.science/r/KGE-Calibrator-6CBB/README.md

Table 4: Effect of different calibration methods on the performance of various KGE models across multiple data sets. For ECE, ACE, and NLL, lower values indicate better calibration performance.

ECE	***TransE***				***ComplEx***				***DistMult***				***RotatE***				***Average***
	WN18	*WN18RR*	*FB15K*	*FB15K-237*	*WN18*	*WN18RR*	*FB15K*	*FB15K-237*	*WN18*	*WN18RR*	*FB15K*	*FB15K-237*	*WN18*	*WN18RR*	*FB15K*	*FB15K-237*	
Uncal	0.502	0.265	0.580	0.212	0.852	0.424	0.696	0.228	0.528	0.389	0.694	0.221	0.429	0.385	0.684	0.224	0.457
PS	0.634	0.031	0.530	0.218	0.854	0.427	0.701	0.229	0.529	0.394	0.700	0.222	0.876	0.425	0.722	0.235	0.483
VS	0.706	0.014	0.646	0.231	0.852	0.424	0.697	0.228	0.528	0.389	0.695	0.215	0.944	0.413	0.739	0.239	0.498
TS	0.634	0.031	0.680	0.203	0.852	0.424	0.701	0.228	0.528	0.389	0.700	0.221	0.687	0.384	0.722	0.223	0.475
PTS	0.523	0.013	0.530	0.231	0.854	0.430	0.060	0.214	0.456	0.393	0.526	0.778	0.337	0.425	0.221	0.365	0.397
KGEC	0.611	0.196	0.408	0.199	0.824	0.377	0.689	0.161	0.501	0.388	0.683	0.165	0.813	0.327	0.642	0.215	0.450
KGEC+	0.171	0.280	0.468	0.150	0.838	0.418	0.678	0.189	0.446	0.383	0.683	0.178	0.467	0.307	0.465	0.094	**0.388**

ACE	***TransE***				***ComplEx***				***DistMult***				***RotatE***				***Average***
	WN18	*WN18RR*	*FB15K*	*FB15K-237*	*WN18*	*WN18RR*	*FB15K*	*FB15K-237*	*WN18*	*WN18RR*	*FB15K*	*FB15K-237*	*WN18*	*WN18RR*	*FB15K*	*FB15K-237*	
Uncal	0.506	0.274	0.565	0.180	0.852	0.424	0.696	0.228	0.528	0.389	0.694	0.220	0.429	0.385	0.684	0.224	0.455
PS	0.628	0.033	0.530	0.217	0.854	0.427	0.701	0.229	0.529	0.394	0.700	0.222	0.876	0.425	0.722	0.235	0.483
VS	0.506	0.274	0.565	0.180	0.852	0.424	0.697	0.228	0.528	0.389	0.694	0.215	0.429	0.385	0.684	0.224	0.455
TS	0.628	0.033	3.312	0.154	0.852	0.423	0.701	0.228	0.528	0.389	0.700	0.220	0.687	0.384	0.722	0.222	0.636
PTS	0.516	0.013	0.530	0.231	0.854	0.424	0.060	0.207	0.446	0.391	0.522	0.778	0.337	0.418	0.221	0.363	0.394
KGEC	0.510	0.283	7.651	0.943	0.823	0.350	0.670	0.161	0.501	0.388	0.666	0.163	0.400	0.278	3.092	0.308	1.074
KGEC+	0.131	0.277	0.298	0.082	0.837	0.418	0.465	0.207	0.457	0.383	0.516	0.199	0.467	0.306	0.465	0.063	**0.348**

NLL	***TransE***				***ComplEx***				***DistMult***				***RotatE***				***Average***
	WN18	*WN18RR*	*FB15K*	*FB15K-237*	*WN18*	*WN18RR*	*FB15K*	*FB15K-237*	*WN18*	*WN18RR*	*FB15K*	*FB15K-237*	*WN18*	*WN18RR*	*FB15K*	*FB15K-237*	
Uncal	2.891	6.582	3.911	5.396	6.892	7.815	5.954	7.513	7.447	7.858	5.919	7.705	1.376	6.145	4.090	5.750	5.828
PS	3.839	7.304	3.829	5.836	8.831	8.974	7.093	8.438	9.117	9.065	7.257	8.621	3.350	7.364	4.799	6.271	6.874
VS	/	/	/	/	6.892	7.814	5.952	7.510	7.446	7.857	5.916	7.692	1.376	/	/	/	6.495
TS	3.839	7.304	1.285	4.909	6.892	7.802	7.093	7.513	7.447	7.856	7.257	7.704	2.069	6.121	4.799	5.617	5.969
PTS	/	9.181	3.829	9.448	9.314	9.171	1.906	5.714	/	9.496	4.847	/	/	/	/	/	6.990
KGEC	2.831	6.330	0.687	4.093	4.856	7.636	6.732	3.811	5.407	7.772	6.444	3.950	1.308	6.327	5.014	6.156	4.960
KGEC+	2.462	5.965	2.529	2.889	4.608	7.010	1.357	2.911	2.843	7.096	1.319	3.106	1.036	4.698	2.031	2.743	**3.413**

likely due to their limited expressive power, as these methods rely on simple parameterizations that lack the flexibility needed for effective probability calibration. (2) Improved performance with advanced baselines: The two advanced baselines (PTS and KGEC) achieve significantly better results than the simple baselines. To be specific, PTS enhances calibration by generalizing TS by computing prediction-specific temperatures, parameterized by a neural network. KGEC exceeds PTS in terms of ECE and NLL by leveraging the Jump Selection Strategy to identify representative data and Multi-Binning Scaling to determine multi-scale bins. However, its performance on ACE is limited due to its reliance on KL divergence for optimization. (3) Superior performance of KGEC+: Compared to all baselines, KGEC+ achieves the best results by integrating the Wasserstein distance for optimization. This modification enhances the calibration process, leading to more reliable probability estimates. These results validate the effectiveness of KGEC+ as an advanced calibration method for KGE models in entity link prediction.

5.4 Efficiency Study for RQ3

To evaluate the efficiency of our proposed method, we compare the training time of KGEC and KGEC+. For a fair comparison, all methods are trained using only a CPU.

Key Observations from Table 5 in Appendix: (1) KGEC+ is the fastest model to train, outperforming all other baseline methods in terms of efficiency. (2) VS and TS have slightly longer training times than KGEC+, primarily due to their simple model structures. (3) PTS, despite achieving strong calibration performance, requires significantly longer training time, which may hinder its practicality in real-world applications. (4) KGEC achieves a reasonable training speed, largely benefiting from the Jump Selection Strategy, which reduces the amount of processed data. However, its reliance on KL divergence still limits its efficiency. (5) KGEC+ significantly accelerates training compared to KGEC by replacing KL divergence with the more expressive and computationally efficient Wasserstein distance. Overall, these results highlight that KGEC+ effectively addresses the inefficiencies of KL divergence, significantly reducing training time while maintaining superior calibration performance.

6 Conclusion

In this paper, we addressed the critical yet often overlooked issue of probability calibration in knowledge graph embedding (KGE) models. We introduced KGE Calibrator+, the first probability calibration method specifically designed for KGE models. Extensive experiments across multiple benchmark datasets demonstrated that KGE Calibrator+ consistently outperforms existing calibration methods, achieving superior performance while maintaining computational efficiency. In future work, we aim to extend KGE Calibrator+ to support more complex multi-relational KGs and explore its applicability to dynamic knowledge graphs.

Limitations

While KGE Calibrator+ demonstrates strong performance in probability calibration for KGE models, several limitations remain: (1) Dependence on Training Data Quality: The effectiveness of Jump Selection Strategy and Multi-Binning Scaling relies on the quality and diversity of training data. If the dataset is highly imbalanced or lacks sufficient representative samples, the calibration performance may degrade. (2) Fixed Calibration Across Different Tasks: Our method is optimized for entity link prediction in static knowledge graphs. However, dynamic KGs and other KGE-based tasks, such as knowledge reasoning and fact verification, may require task-specific modifications to the calibration strategy. (3) Generalization to Other KGE Models: While KGEC+ has been validated on several popular KGE models (e.g., TransE, RotatE), its performance across more complex architectures (e.g., hyperbolic embeddings or transformer-based KGE models) remains an open question. Future research should investigate how KGEC+ can be adapted to these settings.

References

Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. 2019. Uncertainty-based continual learning with adaptive regularization. *Advances in neural information processing systems*, 32.

Jarosław Błasiok, Parikshit Gopalan, Lunjia Hu, and Preetum Nakkiran. 2023. A unifying theory of distance from calibration. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1727–1740.

Jarosław Błasiok and Preetum Nakkiran. 2024. Smooth ECE: Principled reliability diagrams via kernel smoothing. In *ICLR*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, volume 26.

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.

Mohamed H Gad-Elrab, Daria Stepanova, Trung-Kien Tran, Heike Adel, and Gerhard Weikum. 2020. Excut: Explainable embedding-based clustering over knowledge graphs. In *ISWC*, pages 218–237. Springer.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *ICML*, pages 1321–1330. PMLR.

Meelis Kull, Telmo Silva Filho, and Peter Flach. 2017. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial intelligence and statistics*, pages 623–631. PMLR.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Chris Bizer. 2015. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195.

Lihui Liu, Boxin Du, Yi Ren Fung, Heng Ji, Jiejun Xu, and Hanghang Tong. 2021. Kompare: a knowledge graph comparative reasoning system. In *SIGKDD*, pages 3308–3318.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Xingchen Ma and Matthew B Blaschko. 2021. Metacal: Well-controlled post-hoc calibration by ranking. In *International Conference on Machine Learning*, pages 7235–7245. PMLR.

Xianghui Meng, Yang Song, Qingchun Bai, and Taoyi Wang. 2023. Cbki: A confidence-based knowledge integration framework for multi-choice machine reading comprehension. *Knowledge-Based Systems*, 277:110796.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

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

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. 2019. Measuring calibration in deep learning. In *CVPR workshops*, volume 2.

Pouya Pezeshkpour, Yifan Tian, and Sameer Singh. 2020. Revisiting evaluation of knowledge base completion models. *AKBC*.

John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Aishwarya Rao, Narayanan Asuri Krishnan, and Carlos R Rivero. 2024. Using model calibration to evaluate link prediction in knowledge graphs. In *WWW*, pages 2042–2051.

ZAishwarya Rao. 2021. Calibrating knowledge graphs. In *Rochester Institute of Technology*.

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

Tao Shen, Xiubo Geng, Tao Qin, Daya Guo, Duyu Tang, Nan Duan, Guodong Long, and Daxin Jiang. 2019. Multi-task learning for conversational question answering over a large-scale knowledge base. In *EMNLP-IJCNLP*, pages 2442–2451.

Wei Shen, Yang Yang, and Yinan Liu. 2022. Multi-view clustering for open knowledge base canonicalization. In *SIGKDD*, pages 1578–1588.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*, pages 697–706.

Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, volume 18.

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

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

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in neural information processing systems*, 32.

Sudhanshu Tiwari, Iti Bansal, and Carlos R Rivero. 2021. Revisiting the evaluation protocol of knowledge graph completion methods for link prediction. In *Proceedings of the Web Conference 2021*, pages 809–820.

Christian Tomani, Daniel Cremers, and Florian Buettner. 2022. Parameterized temperature scaling for boosting the expressive power in post-hoc uncertainty calibration. In *European Conference on Computer Vision*, pages 555–569. Springer.

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 *ICML*, pages 2071–2080.

Kai Wang, Yu Liu, and Quan Z Sheng. 2021. Neighborhood intervention consistency: Measuring confidence for knowledge graph link prediction. In *IJCAI*, pages 2090–2096.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *WWW*, pages 1271–1279.

Yao Xu, Shizhu He, Li Cai, Kang Liu, and Jun Zhao. 2023. Prediction and calibration: Complex reasoning over knowledge graph with bi-directional directed acyclic graph neural network. In *ACL Findings*, pages 7189–7198.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *ACL*, pages 2346–2357.

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

Qichen Ye, Bowen Cao, Nuo Chen, Weiyuan Xu, and Yuexian Zou. 2023. Fits: Fine-grained two-stage training for knowledge-aware question answering. In *AAAI*, volume 37, pages 13914–13922.

Bianca Zadrozny and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, pages 609–616.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *SIGKDD*, pages 694–699.

Xiangxiang Zeng, Xinqi Tu, Yuansheng Liu, Xiangzheng Fu, and Yansen Su. 2022. Toward better drug discovery with knowledge graph. *Current opinion in structural biology*, 72:114–126.

Ruiqi Zhu, Fangrong Wang, Alan Bundy, Xue Li, Kwabena Nuamah, Lei Xu, Stefano Mauceri, and Jeff Z Pan. 2022. A closer look at probability calibration of knowledge graph embedding. In *IJCKG*, pages 104–109.

A Appendix

A.1 Proof of KL Divergence with t^2

Given:

$$D_{KL}\left(\frac{p(x)}{t^2}, q(x)\right) = \int \frac{p(x)}{t^2} \log\left(\frac{\frac{p(x)}{t^2}}{q(x)}\right) dx$$

We Distribute $\frac{p(x)}{t^2}$ across the Log Terms and separate the terms in the integral:

$$D_{KL}\left(\frac{p(x)}{t^2}, q(x)\right) = \int \frac{p(x)}{t^2} \log\left(\frac{p(x)}{q(x)}\right) dx - \int \frac{p(x)}{t^2} \log(t^2)\, dx \tag{10}$$

Then, we simplify each integral. For the first integral, we can rewrite the first integral by factoring out $\frac{1}{t^2}$:

$$\int \frac{p(x)}{t^2} \log\left(\frac{p(x)}{q(x)}\right) dx = \frac{1}{t^2} D_{KL}(p(x), q(x))$$

For the second integral, since $\log(t^2)$ is constant, we have:

$$\int \frac{p(x)}{t^2} \log(t^2)\, dx = \frac{\log(t^2)}{t^2} \int p(x)\, dx = \frac{\log(t^2)}{t^2}$$

since $\int p(x)\, dx = 1$.

Finally, we combine both integrals, and get:

$$D_{KL}\left(\frac{p(x)}{t^2}, q(x)\right) = \frac{1}{t^2} D_{KL}(p(x), q(x)) - \frac{\log(t^2)}{t^2}$$

For Case 1: $0 < t^2 < 1$

(1) Effect on $\frac{1}{t^2} D_{KL}(p(x), q(x))$: Since $\frac{1}{t^2} > 1$ when $0 < t^2 < 1$, this term increases the overall value of $D_{KL}\left(\frac{p(x)}{t^2}, q(x)\right)$.

(2) Effect on $-\frac{\log(t^2)}{t^2}$: $\log(t^2)$ is negative when $0 < t^2 < 1$, so $-\frac{\log(t^2)}{t^2}$ is positive. Thus, this term further increases $D_{KL}\left(\frac{p(x)}{t^2}, q(x)\right)$.

Summary: When $0 < t^2 < 1$, the entire expression is amplified, leading to a larger divergence than the standard $D_{KL}(p(x), q(x))$.

For Case 2: $t^2 > 1$

(1) Effect on $\frac{1}{t^2} D_{KL}(p(x), q(x))$: When $t^2 > 1$, $\frac{1}{t^2} < 1$, which reduces the contribution of $D_{KL}(p(x), q(x))$ to the overall divergence.

(2) Effect on $-\frac{\log(t^2)}{t^2}$: In this case, $\log(t^2)$ is positive, so $-\frac{\log(t^2)}{t^2}$ is negative, reducing the overall divergence further.

Summary: When $t^2 > 1$, both terms reduce the value, leading to a **smaller divergence** compared to $D_{KL}(p(x), q(x))$.

Conclusion

Scaling $p(x)$ by $\frac{1}{t^2}$ where $t^2 < 1$ emphasizes the divergence, potentially making discrepancies between $p(x)$ and $q(x)$ more significant. Scaling $p(x)$ by $\frac{1}{t^2}$ where $t^2 > 1$ diminishes the divergence, softening the effect of differences between $p(x)$ and $q(x)$.

B Handling Zero Probabilities in Kullback–Leibler Divergence

Let p and q be two discrete probability distributions defined over a finite set $\mathcal{X}$. The Kullback–Leibler divergence from q to p is defined as

$$D_{\mathrm{KL}}(p \parallel q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}.$$

In the computation of $D_{\mathrm{KL}}(p \parallel q)$, special care must be taken for the terms where either $p(x) = 0$ or $q(x) = 0$.

Case 1: $p(x) = 0$

For any $x \in \mathcal{X}$ such that $p(x) = 0$, the corresponding term in the divergence is

$$0 \cdot \log \frac{0}{q(x)}.$$

Although $\log 0$ is undefined, we define this term by considering the limit:

$$\lim_{p(x) \to 0} p(x) \log \frac{p(x)}{q(x)} = 0,$$

since it is well known that

$$\lim_{u \to 0} u \log u = 0.$$

Thus, we adopt the convention

$$0 \cdot \log \frac{0}{q(x)} = 0.$$

Case 2: $q(x) = 0$ **and** $p(x) > 0$

If there exists an $x \in \mathcal{X}$ for which $p(x) > 0$ but $q(x) = 0$, then the term becomes

$$p(x) \log \frac{p(x)}{0}.$$

Since

$$\log \frac{p(x)}{0} = +\infty,$$

Table 5: Training time in seconds taken to calibrate entity link prediction using different methods. Best and second-ranked results are in bold and underlined, respectively. For fair comparison, these results are obtained using CPU only.

Method	*TransE*				*ComplEx*				*DistMult*				*RotatE*				*Average*
	WN18	*WN18RR*	*FB15K*	*FB15K-237*	*WN18*	*WN18RR*	*FB15K*	*FB15K-237*	*WN18*	*WN18RR*	*FB15K*	*FB15K-237*	*WN18*	*WN18RR*	*FB15K*	*FB15K-237*	
PS	50551.471	32130.612	66566.552	22756.968	44484.280	27740.023	66631.859	20060.975	48902.412	31739.057	58074.230	21682.032	46162.422	30198.810	65506.688	20522.725	40856.945
VS	2.857	1.893	25.357	3.493	2.661	1.620	16.228	3.218	4.114	1.914	20.779	3.456	2.656	1.706	25.995	3.277	<u>7.577</u>
TS	5.235	3.207	20.037	6.475	5.063	3.121	18.825	6.276	5.180	3.204	19.734	6.412	5.456	3.171	20.646	6.345	8.649
PTS	3452.440	2123.849	16769.166	5856.000	3432.436	2122.273	16510.019	5764.345	3450.331	2120.555	16898.528	5868.468	3425.148	2113.001	16802.984	5853.287	7035.177
KGEC	2.269	1.334	8.778	3.049	2.295	1.391	8.738	2.950	2.267	1.395	8.914	3.013	2.911	1.331	5716.508	2.996	360.634
KGEC+	2.389	1.429	9.373	3.250	2.350	1.431	9.348	3.205	2.371	1.488	9.349	3.231	2.367	1.426	9.423	3.207	**4.102**

Table 6: Summary table for calibration method used by related works.

Calibration method	Parametric method	Used works
Isotonic Regression (Zadrozny and Elkan, 2002)	No	(Tabacof and Costabello, 2020), (Wang et al., 2021), (Zhu et al., 2022)
Histogram Binning (Zadrozny and Elkan, 2001)	No	(Zhu et al., 2022)
Beta Calibration (Kull et al., 2017)	Yes	(Zhu et al., 2022)
Platt Scaling (Platt et al., 1999)	Yes	(Tabacof and Costabello, 2020), (Wang et al., 2021), (Zhu et al., 2022)
Matrix Scaling (Guo et al., 2017)	Yes	(Safavi et al., 2020)
Vector Scaling (Guo et al., 2017)	Yes	(Safavi et al., 2020)
Temperature Scaling (Guo et al., 2017)	Yes	(Zhu et al., 2022)

the term diverges to $+\infty$. Consequently, the divergence is defined as

$$D_{\mathrm{KL}}(p \parallel q) = +\infty,$$

if there exists any $x \in \mathcal{X}$ with $p(x) > 0$ and $q(x) = 0$.

Summary

Thus, the KL divergence is formally defined as

$$D_{\mathrm{KL}}(p \parallel q) = \begin{cases} \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}, & \text{if } q(x) > 0, \\ +\infty, & \text{otherwise.} \end{cases}$$

This definition ensures that the divergence is finite only when the support of p is a subset of the support of q, and it penalizes models that assign zero probability to events observed under p.

B.1 More Data for Accuracy Affection Study to RQ1

For more results of other KGE models across various datasets after applying different calibration methods in Section 5.2, you can find it at here [6].

[6] https://anonymous.4open.science/r/KGE-Calibrator-6CBB/README.md