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, current 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 (KGEC), the first probability calibration method tailored for KGE models to enhance the trustworthiness of their predictions. To achieve this, we introduce a Jump Selection Strategy that improves efficiency by selecting the most informative instances while filtering out less significant ones. We also propose Multi-Binning Scaling, which models different probability levels separately to increase the model's capacity and flexibility. Additionally, we propose a Wasserstein distance-based loss function to further boost calibration performance. Extensive experiments across multiple datasets demonstrate that KGEC 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.

Query: (Greece, _member_of_domain_region, ?)	
True answer: sibyl	
Ranked candidate entities	Uncalibrated scores
Greece	-0.1873
Holy_See	-0.2946
sibyl	-0.5992
Colosseum	-0.8017
Sistine_Chapel	-0.8683

Figure 1: A real example from the WN18RR (Dettmers et al., 2018) dataset, where ranked candidate entities and their corresponding uncalibrated scores are produced by TransE (Bordes et al., 2013) model. In this example, the correct tail entity "sibyl" is ranked third, demonstrating that existing KGE models perform well under ranking metrics. However, all predicted scores are negative, indicating a lack of trustworthy probability estimates and highlighting the need for probability calibration in KGE models.

However, most KGs are incomplete due to extraction errors and limited input resources. This makes link prediction, also known as knowledge graph completion, crucial for inferring missing links and improving KG quality. To this end, knowledge graph embedding (KGE) models such as TransE (Bordes et al., 2013) and ComplEx (Trouillon et al., 2016) tackle this problem by learning latent representations of entities and relations to score the plausibility of candidate triples. Beyond link prediction, KGE models have demonstrated remarkable success across diverse applications, including entity alignment (Sun et al., 2018) and canonicalization (Shen et al., 2022).

While the accuracy of KGE models has seen significant advancements, the critical issue of probability calibration remains largely overlooked. Specifi-

cally, KGE models should output calibrated probabilities alongside their predictions. However, they typically produce uncalibrated scores (Pezeshkpour et al., 2020; Tabacof and Costabello, 2020). This stems from link prediction being framed as a ranking task, where metrics like HITS@N and Mean Rank (MR) prioritize relative ordering while ignoring the reliability of output scores. As a result, models can assign implausible scores to correct entities yet still perform well, as shown in Figure 1. Such limitations hinder their use in high-stakes domains such as drug and protein target discovery (Zeng et al., 2022; Mohamed et al., 2020), where trustworthy probabilities are essential.

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 into well-calibrated probability estimates. As a post-processing technique, calibration improves the trustworthiness of link prediction results, making them more reliable for downstream applications. However, probability calibration in KGE poses unique challenges compared to traditional classification. Image classification datasets like CIFAR-100 (Krizhevsky et al., 2009) or document classification datasets like SST (Socher et al., 2013) involve tens or hundreds of classes. In contrast, KGE tasks treat each entity as a distinct class, leading to massive class spaces (e.g., FB15K and WN18 have 14,951 and 40,943 entities, respectively). This high cardinality leads to tiny per-class probabilities and makes the calibration process extremely sensitive. Even minor perturbations can distort the original ranking and negatively impact link prediction performance. Therefore, preserving the original ranking quality becomes a critical requirement, posing a distinctive challenge for probability calibration in the KGE setting.

Despite its importance and unique challenges, probability calibration in KGE remains largely underexplored. Prior studies (Tabacof and Costabello, 2020; Pezeshkpour et al., 2020) have shown that popular KGE models produce poorly calibrated scores, resulting in unreliable probability estimates. Several off-the-shelf calibration methods, such as Platt Scaling, Isotonic Regression, and Temperature Scaling, have been evaluated (Safavi et al., 2020; Zhu et al., 2022), but these methods are designed for standard classifiers and are not well-suited to the scale and ranking-sensitive nature of KGE. A few works have explored calibration in specific tasks, including triple classification (Tabacof and Costabello, 2020), relation prediction (Safavi et al., 2020), and low-dimensional entity expit transformations (Wang et al., 2021). However, no existing approach offers a calibration method explicitly tailored to the probabilistic characteristics of KGE models. This leaves a critical gap in improving the trustworthiness of KGE-based link prediction.

To fill this gap, we propose KGE Calibrator (KGEC), the first probability calibration method tailored specifically for KGE models. To enhance training efficiency under the large-scale class space characteristic of KGE, we introduce the Jump Selection Strategy, which selects the most informative instances while discarding less significant ones. To increase model expressiveness and better captures the ranking-sensitive nature of KGE predictions, we propose Multi-Binning Scaling, which models different probability levels separately, thereby increasing model capacity and flexibility. Additionally, we propose a Wasserstein distance-based loss function to further boost calibration performance. To the best of our knowledge, this is the first use of the Wasserstein distance for probability calibration.

Contributions. Our major contributions can be summarized as follows:

• We evaluate nine widely used post-processing calibration methods and find that four of them are unsuitable for entity 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, which addresses the challenge of large class space in calibration while preserving the original ranking performance.

• A thorough experimental study over four datasets demonstrates that KGEC consistently outperforms existing calibration methods in both performance and efficiency.

2 Related Work

Probability Calibration in KGE Models. Several studies have highlighted the lack of well-calibrated probability estimates in KGE models. Early work by (Tabacof and Costabello, 2020) and (Pezeshkpour et al., 2020) showed that widely used KGE models are poorly calibrated in triple classification tasks. To address this, (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 relation prediction. A broader evaluation by (Zhu et al., 2022) tested additional off-the-shelf calibration techniques, including Histogram Binning (Zadrozny and Elkan, 2001), Beta Calibration (Kull et al., 2017), and Temperature Scaling (Guo et al., 2017) for triple classification. Furthermore, (Rao, 2021) examined calibration under both closed-world and open-world assumptions. While these works shed light on the calibration issue in KGE, they all rely on existing techniques originally designed for traditional classification problems. None propose a calibration method specifically tailored for KGE models, leaving a critical gap in the literature.

Expit Transformations. Expit transformations aim to convert uncalibrated scores into probabilities using functions such as the Sigmoid (Nickel et al., 2015; Tabacof and Costabello, 2020; Zhu et al., 2022) and Softmax (Pezeshkpour et al., 2020). More approaches include neighborhood intervention consistency (NIC) (Wang et al., 2021) and min-max scaling (Rao, 2021). However, recent research has shown that even when expit-transformed scores can be interpreted as probabilities, they are still uncalibrated and unreliable (Zhu et al., 2022). As a result, these expit transformations are generally viewed as a preliminary step, typically followed by a dedicated calibration method such as Platt Scaling or Isotonic Regression. In fact, (Zhu et al., 2022) concluded that expit transformations are ineffective in most cases and suggested probability calibration as a better approach. Following this direction, our work focuses exclusively on probability calibration and does not include expit transformations as part of our method design.

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 3 in Appendix A lists the score functions of the most popular KGE models.

3.3 Link Prediction

Link prediction, the primary task for KGE models, includes entity and relation prediction. Entity prediction is more challenging due to the large number of candidate entities. For example, WN18RR (Dettmers et al., 2018) contains 40,943 entities but only 11 relations. This paper focuses on the more difficult entity prediction task. To be specific, the entity prediction includes head and tail prediction. For head 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

In this section, we present our proposed method, KGE Calibrator (KGEC). We begin with the introduction of our proposed Jump Selection Strategy and Multi-Binning Scaling, thereafter describe the Wasserstein distance-based loss function subsequently.

4.1 Jump Selection Strategy

To improve training efficiency in the context of large-scale class spaces inherent to KGE tasks, it is crucial to focus on the most informative instances while discarding less significant one during training the calibration method. Inspired by (Shen et al., 2022), we propose the Jump Selection Strategy, which selects the most significant instances for training rather than using all available instances. This Jump Selection Strategy is summarized in Algorithm 1, and we elaborate it as follows.

Given a query set $Q = \{q_1, ..., q_i, ..., q_N\}$ where $q_i = (?, r_i, t_i)$, and a set of candidate entities $\mathcal{E} = \{e_1, ..., e_i, ..., e_M\}$, we first generate candidate triples $\xi_{ij} = (e_j, r_i, t_i)$ for all i and j (line 1

Algorithm 1 Jump Selection Strategy

Input: A set of queries $Q = \{q_1, ..., q_i, ..., q_N\}$, where $q_i = (?, r_i, t_i)$ for $i = 1, \ldots, N$, a set candidate entities $\mathcal{E} = \{e_1, ..., e_j, ..., e_M\}$ for $j = 1, \ldots, M$, a trained KGE model ψ

1: **Generate** candidate triples for each query: $\xi_{ij} \leftarrow (e_j, r_i, t_i)$
2: **Compute** uncalibrated scores for each query: $x_{ij} \leftarrow \psi(\xi_{ij}) = \psi(\mathbf{e}_j, \mathbf{r}_i, \mathbf{t}_i)$
3: Form the uncalibrated scores into a score vector: $\mathbf{X}_i \leftarrow \{x_{i1}, ..., x_{ij}, ..., x_{iM}\}$
4: **Compute** probabilities: $\mathbf{P}_i \leftarrow \sigma_{\text{SM}}(\mathbf{X}_i)$ for each query
5: Form the uncalibrated probabilities into a probability matrix: $\mathbf{P} \leftarrow \{\mathbf{P}_1, ..., \mathbf{P}_i, ..., \mathbf{P}_N\}^\top$
6: **Sort** $\mathbf{P}$ in descending order by row to obtain $\tilde{\mathbf{P}}$
7: **for** $j = 1$ **to** $M - 1$ **do**
8: $\quad J_j \leftarrow D_{KL}(\tilde{\mathbf{P}}_j \,\|\, \tilde{\mathbf{P}}_{j+1})$
9: **end for**
10: $J^* \leftarrow \arg\max_j J_j$
11: $p^* \leftarrow \tilde{\mathbf{P}}_{J^*}$

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

in Algorithm 1). The KGE model ψ is then used to compute uncalibrated scores $x_{ij} = \psi(\xi_{ij})$ for each candidate triple, forming a score vector $\mathbf{X}_i$ for each query q_i (lines 2 – 3 in Algorithm 1). These scores are transformed into probability vectors $\mathbf{P}_i$ via the Softmax function σ_{SM}, and assembled into a probability matrix $\mathbf{P}$ (lines 4 – 5 in Algorithm 1). We then sort each row of $\mathbf{P}$ in descending order to obtain $\tilde{\mathbf{P}}$ (line 6 in Algorithm 1), so that higher probabilities appear first. To identify the most informative instance, we compute the Jump Measure J_j for each adjacent column pair in the sorted probability matrix $\tilde{\mathbf{P}}$ using KL divergence (lines 7 – 8 in Algorithm 1). The index J^* corresponding to the maximum Jump Measure is then selected, and its associated probability vector $p^* = \tilde{\mathbf{P}}_{J^*}$ is used as the most informative sample for subsequent calibration training (lines 9 – 11 in Algorithm 1).

4.2 Multi-Binning Scaling

Temperature Scaling (TS) (Guo et al., 2017) is a widely used post-hoc calibration method due to its simplicity and its ability to preserve the original model's ranking order, which is an essential property in the KGE link prediction task. TS achieves this by scaling the uncalibrated probabilities using a single scalar temperature parameter $T > 0$, thereby maintaining the relative ordering of scores. However, TS suffers from limited expressiveness, as it applies the same transformation regardless of the input probability magnitude (e.g., both 0.1 and 0.9 are scaled identically), which can lead to suboptimal calibration performance.

To address this limitation, we introduce Multi-Binning Scaling (MBS), a more flexible approach that maintains the ranking-preserving property of TS while improving calibration quality. Inspired by histogram binning (Zadrozny and Elkan, 2001), we partition the uncalibrated probabilities $p^* = \{p_1^*, ..., p_i^*, ..., p_N^*\}$ into W mutually exclusive bins $B_1, \ldots, B_w, ..., B_W$. Each bin is associated with an independent scalar temperature parameter T_w. Uncalibrated probabilities falling into bin B_w are calibrated using:

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

where σ_{SM} denotes the Softmax function. The squared temperature form follows convention in temperature-based calibration and provides smoother gradient behavior.

To define bin boundaries, we divide the interval $[0, 1]$ into W equal-length segments:

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

so that bin B_w corresponds to the interval $(a_w, a_{w+1}]$. In this paper, we adopt uniformly divided equal-length intervals for bin boundaries to maintain simplicity. More advanced strategies, such as adaptive bin boundaries, are left for future exploration.

For probability vectors $p^- = \tilde{\mathbf{P}}_j$ where $j \neq J^*$ (i.e., those not selected by Jump Selection Strategy), we reuse the temperature parameter T_m associated with p^* to calibrate them. This design avoids the overhead of rebinning and retraining temperature parameters, while ensuring that the original ranking produced by the model remains unaffected. Overall, MBS combines the ranking-preserving property of TS with the expressiveness of bin-based transformations, enabling more accurate and robust calibration for KGE link prediction without compromising original ranking performance.

4.3 Optimization

While Kullback–Leibler (KL) divergence is a commonly used loss function in deep learning, it poses

notable limitations for probability calibration in KGE models. This is particularly evident in high-cardinality tasks such as entity prediction, where each entity corresponds to a unique class and the class space can include tens of thousands of candidate entities. First, in such large class spaces, the predicted probabilities for most entities are extremely small. When the true label probability q_i is nonzero but the predicted probability p_i approaches zero, the corresponding KL loss becomes negligible. This results in near-zero loss values for many informative instances, reducing their impact during training and weakening the effectiveness of calibration. Second, KL divergence towards infinity when $p_i > 0$ and q_i approaches zero, causing the loss to diverge toward infinity. In practice, this can cause gradient instability or explosion, particularly in sparse or imbalanced prediction scenarios. These issues compromise the robustness and reliability of probability calibration in KGE. A detailed analysis is provided in Appendix B.

To address these issues, we propose using the Wasserstein distance as the loss function for KGEC. Unlike KL divergence, the Wasserstein distance provides a more stable and geometrically meaningful way to compare probability distributions by considering the minimum cost of transforming one distribution into another. This perspective is especially valuable in calibration, where we aim to align uncalibrated scores with true probability distributions while preserving their structure.

The Wasserstein distance models calibration as an optimal transport (OT) problem, where the goal is to find the most efficient way to move probability mass from the uncalibrated distribution p^* to the target distribution q. The feasible set of transport plans is defined by the transportation polytope $U(p^*, q)$, which contains all nonnegative transport matrices P:

$$U(p^*, q) = \{\mathbf{P} \in \mathbb{R}_+^{d\times d} | \mathbf{P}\mathbf{1}_d = p^*, \mathbf{P}^\top \mathbf{1}_d = q\}, \tag{3}$$

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.

$$D_{WD}(p^*, q) = \min_{P\in U(p^*,q)} \langle P, M\rangle = \sum_{m,n} P_{m,n} M_{m,n}, \tag{4}$$

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:

$$D_{SD}(p^*, q) = \left\langle P^\lambda, M \right\rangle, \tag{5}$$

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

$$P^\lambda = \underset{P\in U(p^*,q)}{\arg\min} \langle P, M\rangle - \frac{1}{\lambda} h(P), \tag{6}$$

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

$$\begin{aligned} u^t &= p^* \oslash (K^\top v^{t-1}), \\ v^t &= q \oslash (K u^t), \end{aligned} \tag{7}$$

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:

$$P^\lambda = \mathrm{diag}(v^t) K \,\mathrm{diag}(u^t), \tag{8}$$

This Sinkhorn-regularized Wasserstein loss enables more stable optimization and improves calibration performance, particularly under the large class space settings encountered in KGE tasks.

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 KGEC method surpass the performance of existing methods while preserving the original ranking quality?
- **RQ3**: Is our proposed KGEC 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 ranking results evaluation for **RQ1**. Section 5.3 presents the effectiveness evaluation for **RQ2**. Section 5.4 discusses the training time and memory usage for **RQ3**.

5.1 Experimental Setting

5.1.1 datasets

We evaluate our proposed model on four popular datasets, 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., 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 4 in Appendix C.

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 3. It is noted that any KGE models could be employed as the input of our KGEC 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 this work, we focus exclusively on post-hoc probability calibration methods to preserve the original ranking of KGE models. As such, techniques that modify model training, 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 beyond the scope of this study. Additionally, we fail to report results for Beta Calibration (Kull et al., 2017) due to its extremely high computational cost. For instance, even on the smallest dataset (WN18RR), this method required over 60 hours to complete, rendering it impractical for our large-scale experiments. Lastly, we clarify that this work focuses solely on probability calibration and does not consider expit transformations, such as replacing the Softmax function with Sigmoid or NIC (Wang et al., 2021). These transformations are thus fall outside the scope of our study.

5.1.4 Evaluation measures

Evaluating calibration performance requires both reliable metrics to detect miscalibration and effective techniques to fix such distortion. In this work, we adopt three widely used evaluation metrics: Expected Calibration Error (ECE) (Naeini et al., 2015), Adaptive Calibration Error (ACE) (Nixon et al., 2019), and Negative Log-Likelihood (NLL). Each metric captures different aspects of calibration quality. Due to space constraints, we refer readers to (Naeini et al., 2015; Nixon et al., 2019) for detailed formulations. To give an overall evaluation of each method, we calculate the average of each metric for different dataset 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 used 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 PTS[4] is from their official code. For the hyper-parameter setting of KGEC, the number of bins is set to 10, the learning rate is set to 0.01, the batch size is set to 32, the initial temperature for each bin is set to 1.0 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 datasets. 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 1 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. For more results of other KGE models across various datasets after applying different calibration methods are shown in Table 8, Table 9 and Table 10.

From the experimental results in Table 1, 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 prediction task; (2) KGEC maintains the ranking 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.

Table 1: 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
			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
			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
			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

5.3 Effectiveness Study for RQ2

Table 2 presents the calibration performance of various methods across multiple KGE models and datasets. Notably, baselines such as HB, IR, BBQ, MS, and Meta-Cal are excluded, due to their detrimental impact on ranking performance, as shown in Section 5.2. Since preserving the original ranking order is essential in KGE settings, these calibration methods that degrade ranking performance are considered unsuitable for practical deployment and omitted from further evaluation.

Overall, across all datasets and models, KGEC

[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

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

[5] https://anonymous.4open.science/r/KGE-Calibrator-D780/README.md

Table 2: Effect of different calibration methods on the performance of various KGE models across multiple datasets. 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	0.171	0.280	0.459	0.150	0.838	0.418	0.678	0.189	0.446	0.383	0.683	0.178	0.467	0.307	0.466	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	0.131	0.277	0.293	0.082	0.833	0.418	0.465	0.207	0.457	0.383	0.516	0.199	0.467	0.306	0.466	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	2.462	5.965	2.536	2.889	4.350	6.965	1.357	2.911	2.843	7.119	1.319	3.106	1.036	4.698	2.033	2.743	**3.396**

consistently achieves the lowest average ECE, ACE, and NLL, clearly outperforming all competitive baselines. Key findings from Table 2 include: (1) Limited effectiveness of simple baselines: PS, VS, and TS often perform worse than the uncalibrated models. Their poor performance is likely due to their low model capacity, which is insufficient to capture complex calibration patterns in high-cardinality KGE settings. (2) Improved results with PTS: PTS shows marked improvement over simple baselines by predicting temperature parameters adaptively using a neural network. This flexibility enables better handling of distributional variation, leading to improved calibration performance. (3) Superior performance of KGEC: KGEC achieves the best overall results across all metrics and datasets. Its combination of Jump Selection Strategy, Multi-Binning Scaling, and Wasserstein distance-based loss function effectively addresses the challenges of KGE calibration while preserving ranking quality.

5.4 Efficiency Study for RQ3

Table 5 and Table 6 report the training time and memory usage of different calibration methods across multiple KGE models and datasets. All methods are evaluated on CPU-only environments to ensure fair comparison.

Key Observations from these two tables: (1) KGEC is the most efficient model in both training time and memory usage, consistently outperforming all baseline methods. (2) VS and TS exhibit comparable efficiency, with slightly longer training times than KGEC, which can be attributed to their simple parametric structures. (3) PTS incurs significantly higher computational costs, both in time and memory, despite its strong calibration performance. This high overhead may limit its applicability in large-scale or resource-constrained scenarios. (4) PS is the slowest method, largely due to the immense number of classes in KGE settings, which makes binary logistic regression computationally expensive.

5.5 Ablation Study, Sensitivity Analysis, and Case Study

Due to space limitations, additional experiments, including the ablation study, sensitivity analysis, and case study, are provided in the Appendix. Please refer to Appendix D, E, and F for details.

6 Conclusion

In this paper, we propose KGEC, the first probability calibration method specifically designed for KGE models. KGEC integrates a Wasserstein distance-based loss function, a multi-binning scaling module, and a jump selection strategy to effectively calibrate the predictive probabilities of KGE models without sacrificing ranking performance. Comprehensive experiments across multiple KGE models and benchmark datasets demonstrate that KGEC significantly outperforms existing calibration baselines in terms of effectiveness and efficiency. Overall, KGEC establishes a strong and efficient foundation for trustworthy link prediction. Future work may explore its applicability to dynamic knowledge graphs or integrating it with uncertainty-aware reasoning systems.

Limitations

While KGEC achieves strong performance in calibrating probability estimates for KGE models, several limitations remain:

(1) Limited Exploration of Expit Transformation Functions. In this work, we adopt the Softmax function as the expit transformation, as our primary focus is on the calibration method itself. However, alternative approaches, such as NIC (Wang et al., 2021) and min-max normalization (Rao, 2021), may further improve performance and merit exploration in future work.

(2) Task-Specific Calibration Considerations. KGEC is optimized for static entity prediction tasks in knowledge graphs. Its effectiveness in other KGE-based applications, such as multi-hop reasoning, fact verification, or temporal/dynamic KG settings, remains untested. These tasks may require adaptation or redesign of the calibration strategy to accommodate different data characteristics and evaluation protocols.

(3) Limited Evaluation Across Advanced KGE Architectures. While KGEC has been extensively evaluated on several representative KGE models (e.g., TransE, DistMult, ComplEx, and RotatE), its generalization to more complex architectures, such as hyperbolic embeddings, graph neural networks, or transformer-based KGE models, has not yet been studied. Extending KGEC to these settings poses challenges in modeling and scalability, and is an important direction for future work.

References

Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. 2019. Uncertainty-based continual learning with adaptive regularization. *NeurIPS*, 32.

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 *NeurIPS*, 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 *AAAI*, volume 32.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 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.

Alex Krizhevsky et al. 2009. Learning multiple layers of features from tiny images.

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. *NeurIPS*, 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 *ICLR*.

Xingchen Ma and Matthew B Blaschko. 2021. Meta-cal: Well-controlled post-hoc calibration by ranking. In *ICML*, 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 *AAAI*, volume 29.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

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.

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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642.

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. *NeurIPS*, 32.

Christian Tomani, Daniel Cremers, and Florian Buettner. 2022. Parameterized temperature scaling for boosting the expressive power in post-hoc uncertainty calibration. In *ECCV*, 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.

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 Score functions of popular KGE models

For popular KGE models, we show the score functions of them in Table 3.

Table 3: 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)	$-\|\mathbf{h}\circ\mathbf{r}-\mathbf{t}\|$

B Handling Zero Probabilities in KL Divergence

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

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

While this expression is well-defined when both $p(x) > 0$ and $q(x) > 0$, edge cases involving zero probabilities require special attention. Below, we analyze two important cases.

Case 1: $p(x) = 0$

When $p(x) = 0$, the corresponding term in the summation becomes:

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

Although $\log 0$ is undefined, this term is conventionally set to zero. This is justified by the limit:

$$\lim_{u\to 0^+} u\log\frac{u}{q(x)} = 0.$$

Hence, for numerical stability and analytical consistency, we define:

$$p(x)\log\frac{p(x)}{q(x)} = 0 \quad \text{when } p(x) = 0.$$

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

This case is more problematic. If $p(x) > 0$ and $q(x) = 0$, the logarithmic term becomes:

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

which leads to:

$$p(x)\log\frac{p(x)}{q(x)} = +\infty.$$

Thus, the KL divergence is undefined (i.e., infinite) in this case. Formally:

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

if $\exists x \in \mathcal{X}$ such that $p(x) > 0$ and $q(x) = 0$.

Table 4: Statistics of the used KGE datasets.

dataset	*#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

Summary

Each term $p(x)\log\frac{p(x)}{q(x)}$ in the KL divergence is interpreted as follows:

- If $p(x) = 0$, the term is defined as 0 (by convention via limiting argument).
- If $p(x) > 0$ and $q(x) = 0$, the term is $+\infty$, causing the entire divergence to diverge.

Thus, the KL divergence is finite if and only if the support of p is a subset of the support of q:

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

This behavior makes KL divergence highly sensitive to support mismatch. In high-cardinality tasks such as entity prediction in KGs, sparse output distributions and zero-valued targets frequently occur. This can cause instability during training, particularly in gradient-based optimization, and may compromise calibration performance if not handled appropriately.

C Statistics of the used KGE datasets

The statistics of the used KGE datasets are summarized into Table 4.

D Ablation Study

To assess the individual contribution of each component in KGEC, we perform a comprehensive ablation study across five key metrics: ECE, ACE, NLL, training time, and memory usage. Figure 2 reports the average performance across all datasets and KGE models, providing an overall comparison of model variants. Detailed experimental results for each component on individual datasets and KGE models are presented in Table 11.

We evaluate the following four variants: (1) KGEC: The full model, incorporating all components—Jump Selection Strategy (JSS), Multi-Binning Scaling (MBS), and the Wasserstein distance-based loss. (2) KGEC-loss: Replaces the Wasserstein loss with KL divergence while retaining JSS and MBS. (3) KGEC-loss-MBS: Further

Table 5: Training time in seconds taken to calibrate entity 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	2.727	1.776	10.873	3.602	2.698	1.727	10.560	3.624	2.741	1.696	10.645	3.705	2.662	1.658	10.758	4.003	**4.716**

Table 6: Memory usage in MBs taken to calibrate entity 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	1564.336	950.762	5706.102	1948.508	1566.598	950.270	5706.832	1948.664	1565.820	949.633	5705.828	1947.574	1566.477	950.793	5706.875	1948.371	2542.715
VS	84.477	84.383	86.098	84.348	82.059	83.152	86.918	80.770	83.609	83.883	80.883	81.320	80.570	83.145	86.152	80.941	<u>83.294</u>
TS	1562.625	948.453	5703.750	1947.629	1562.984	949.285	5703.047	1945.566	1562.340	948.504	5704.801	1945.828	1562.914	948.566	5703.359	1944.730	2540.274
PTS	6655.574	7017.359	11154.340	9554.723	6804.816	7022.313	10185.500	9629.871	6957.012	6696.055	10180.105	9407.988	7047.270	7074.051	10521.520	8659.395	8410.493
KGEC	30.484	28.289	7.570	15.273	26.652	32.176	9.535	15.285	34.316	32.047	10.531	13.492	34.320	32.191	7.551	16.930	**21.665**

removes MBS, retaining only JSS and KL divergence. (4) KGEC-loss-MBS-JSS: The base version using only KL divergence, without any of the proposed enhancements.

Key Observations: (1) Full Model Superiority: KGEC achieves the best performance across all five metrics. It yields the lowest calibration errors (ECE = 0.388, ACE = 0.348, NLL = 3.396) while maintaining high efficiency (training time = 4.716s, memory usage = 21.665MB). (2) Impact of Wasserstein Loss: Comparing KGEC to KGEC-loss reveals substantial calibration improvements, validating the advantage of using Wasserstein distance over KL divergence in high-cardinality, ranking-sensitive KGE settings. This supports our hypothesis that the Wasserstein-based objective is better suited to the probability distribution landscape of KGE. (3) Effect of MBS: Removing MBS (KGEC-loss vs. KGEC-loss-MBS) degrades ECE (from 0.450 to 0.487) and NLL (from 4.960 to 5.590), indicating that MBS enhances calibration by modeling probability intervals more effectively. Interestingly, ACE improves after removing MBS. This anomaly may arise because the KL divergence used in KGEC-loss amplifies ACE more than expected, suggesting ACE is especially sensitive to the choice of loss function. (4) Efficiency Gain from JSS: While KGEC-loss-MBS and KGEC-loss-MBS-JSS exhibit similar calibration performance, the inclusion of JSS dramatically reduces training time (from 65.871s to 4.659s) and memory usage (from 97.608MB to 20.032MB), confirming JSS's effectiveness in improving computational efficiency.

Overall, all three components are essential for balancing calibration performance and computational cost. MBS and Wasserstein loss enhance calibration performance, while JSS ensures scalability. The full KGEC model delivers the strongest overall performance.

E Sensitivity Analysis

To assess the robustness and stability of our proposed KGEC method, we conduct a comprehensive sensitivity analysis by varying three critical hyperparameters: the number of bins, the initial temperature, and the learning rate. We evaluate the impact of each parameter on three calibration metrics, i.e., ECE, ACE, and NLL, across all KGE models and datasets. Results are summarized in Tables 12, 13, and 14.

Effect of the Number of Bins. We vary the number of bins from 1 to 20. Table 12 shows that using only one bin (equivalent to vanilla temperature scaling) results in poor performance across all metrics, highlighting its limited flexibility. As the number of bins increases, KGEC becomes more expressive and better calibrated. The best average performance is observed at 19 bins (ECE = 0.352, ACE = 0.343, NLL = 3.361), though results are stable within the 10–20 bin range. This confirms the importance of multi-binning for modeling diverse score distributions, while also indicating that KGEC is robust to bin selection within a reasonable interval.

Effect of Initial Temperature. We examine initial temperature values ranging from 0 to 2.0. As shown in Table 13, extreme initializations (e.g., 0.0 or 2.0) lead to degraded performance due to optimization instability. An initial temperature of 1.0 yields the best results (ECE = 0.388, ACE = 0.348, NLL = 3.396), aligning with standard practice in temperature scaling (Guo et al., 2017). The results

Table 7: 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)

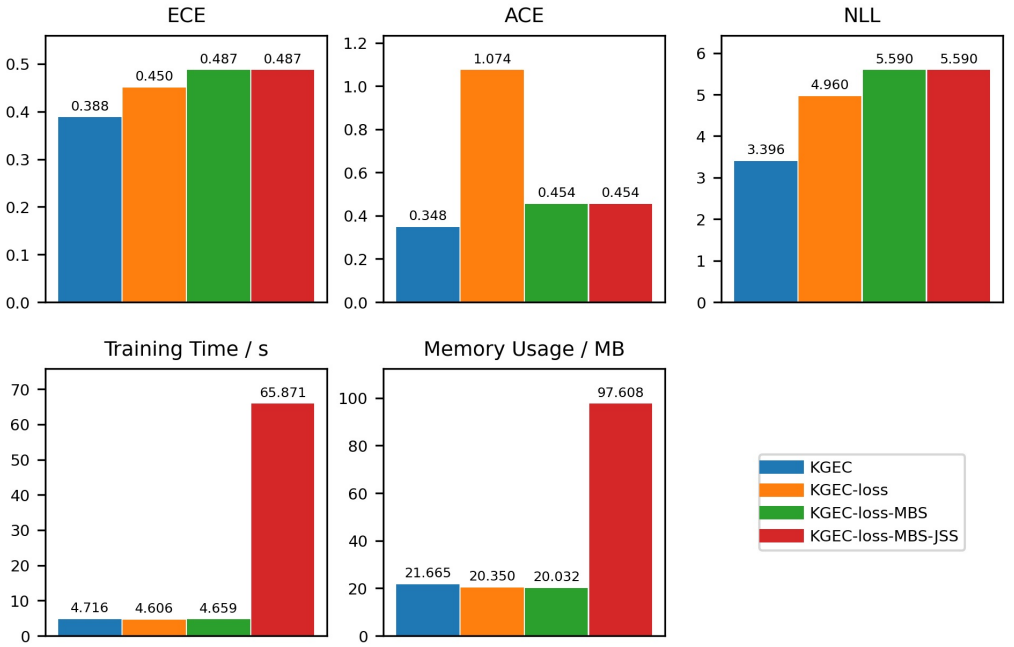

Figure 2: Ablation study of KGEC components across five evaluation metrics: ECE, ACE, NLL, training time (seconds), and memory usage (MB). Lower values indicate better performance.

indicate that KGEC is relatively insensitive to this hyperparameter, as long as it is initialized within a moderate range.

Effect of Learning Rate. Table 14 presents results under learning rates ranging from 0.001 to 0.1. We find that too small learning rates (e.g., 0.001) may underfit the calibration model, while overly large values (e.g., 0.1) can cause instability and degraded performance. The learning rate of 0.01 achieves the best overall calibration (ECE = 0.388, ACE = 0.348, NLL = 3.396), striking a balance between convergence speed and stability.

Summary. Across all experiments, KGEC demonstrates strong robustness to hyperparameter variations. The best performance is consistently achieved with moderate hyperparameter values: a bin count between 10 and 20, an initial temperature near 1.0, and a learning rate around 0.01. These findings suggest that KGEC is both stable and practical, requiring minimal hyperparameter tuning for optimal performance across diverse KGE models and datasets.

Query: (Greece, _member_of_domain_region, ?)
True answer: sibyl

Ranked candidate entities	Uncalibrated scores	Calibrated probabilities
Greece	-0.1873	0.0302
Holy_See	-0.2946	0.0272
sibyl	-0.5992	0.0200
Colosseum	-0.8017	0.0164
Sistine_Chapel	-0.8683	0.0153
Roman	-1.1427	0.0116
Italy	-1.1464	0.0116
Rome	-1.1873	0.0111
Seven_Hills_of_Rome	-1.3174	0.0098
augur	-1.3962	0.0090

Figure 3: Case 1 from the WN18RR dataset using the TransE model.

F Case Study

To illustrate the practical benefits of KGEC calibration, we present two representative case studies from the WN18RR dataset using the TransE

Table 8: Effect of different calibration methods on the performance of the ComplEx model across various datasets.

Method	*MR*	*MRR*	*HITS@1*	*HITS@3*	*HITS@10*
			WN18		
Uncal	311	0.893	0.854	0.925	0.953
PS	311	0.893	0.854	0.925	0.953
HB	14328 ↓	0.274 ↓	0.262 ↓	0.285 ↓	0.289 ↓
IR	14094 ↓	0.290 ↓	0.280 ↓	0.298 ↓	0.304 ↓
BBQ	13657 ↓	0.236 ↓	0.194 ↓	0.271 ↓	0.306 ↓
VS	305 ↑	0.893	0.854	0.925	0.953
MS	16825 ↓	0.011 ↓	0.004 ↓	0.012 ↓	0.022 ↓
TS	311	0.893	0.854	0.925	0.953
Meta-Cal	1260 ↓	0.851 ↓	0.813 ↓	0.880 ↓	0.908 ↓
PTS	311	0.893	0.854	0.925	0.953
KGEC	311	0.893	0.854	0.925	0.953
			WN18RR		
Uncal	5469	0.469	0.428	0.486	0.552
PS	5469	0.469	0.428	0.486	0.552
HB	18836 ↓	0.107 ↓	0.100 ↓	0.112 ↓	0.118 ↓
IR	18244 ↓	0.103 ↓	0.090 ↓	0.110 ↓	0.124 ↓
BBQ	18200 ↓	0.087 ↓	0.076 ↓	0.097 ↓	0.105 ↓
VS	5447 ↑	0.469	0.428	0.486	0.552
MS	18191 ↓	0.009 ↓	0.003 ↓	0.009 ↓	0.020 ↓
TS	5469	0.469	0.428	0.486	0.552
Meta-Cal	6416 ↓	0.445 ↓	0.407 ↓	0.459 ↓	0.522
PTS	5469	0.469	0.428	0.486	0.552
KGEC	5469	0.469	0.428	0.486	0.552
			FB15K		
Uncal	45	0.770	0.703	0.816	0.885
PS	45	0.770	0.703	0.816	0.885
HB	1747 ↓	0.610 ↓	0.543 ↓	0.661 ↓	0.724 ↓
IR	970 ↓	0.652 ↓	0.579 ↓	0.704 ↓	0.780 ↓
BBQ	797 ↓	0.597 ↓	0.509 ↓	0.656 ↓	0.757 ↓
VS	43 ↑	0.770	0.703	0.816	0.886 ↑
MS	3693 ↓	0.025 ↓	0.010 ↓	0.024 ↓	0.055 ↓
TS	45	0.770	0.703	0.816	0.885
Meta-Cal	484 ↓	0.715 ↓	0.651 ↓	0.759 ↓	0.826 ↓
PTS	45	0.770	0.703	0.816	0.885
KGEC	45	0.770	0.703	0.816	0.885
			FB15K-237		
Uncal	166	0.322	0.230	0.352	0.511
PS	166	0.322	0.230	0.352	0.511
HB	2882 ↓	0.274 ↓	0.201 ↓	0.305 ↓	0.420 ↓
IR	2185 ↓	0.296 ↓	0.220 ↑	0.328 ↓	0.449 ↓
BBQ	1661 ↓	0.249 ↓	0.176 ↓	0.273 ↓	0.399 ↓
VS	166	0.322	0.230	0.352	0.512 ↑
MS	3704 ↓	0.033 ↓	0.014 ↓	0.032 ↓	0.070 ↓
TS	166	0.322	0.230	0.352	0.511
Meta-Cal	267 ↓	0.310 ↓	0.218 ↓	0.339 ↓	0.498 ↓
PTS	166	0.322	0.230	0.352	0.511
KGEC	166	0.322	0.230	0.352	0.511

Table 9: Effect of different calibration methods on the performance of the DistMult model across various datasets.

Method	*MR*	*MRR*	*HITS@1*	*HITS@3*	*HITS@10*
			WN18		
Uncal	227	0.685	0.529	0.829	0.933
PS	227	0.685	0.529	0.829	0.933
HB	14718 ↓	0.240 ↓	0.216 ↓	0.262 ↓	0.271 ↓
IR	14271 ↓	0.260 ↓	0.237 ↓	0.279 ↓	0.294 ↓
BBQ	13614 ↓	0.201 ↓	0.154 ↓	0.232 ↓	0.293 ↓
VS	224 ↑	0.685	0.529	0.829	0.933
MS	16984 ↓	0.011 ↓	0.004 ↓	0.012 ↓	0.022 ↓
TS	227	0.685	0.529	0.829	0.933
Meta-Cal	770 ↓	0.663 ↓	0.508 ↓	0.805 ↓	0.908 ↓
PTS	240 ↓	0.685	0.529	0.829	0.932 ↓
KGEC	227	0.685	0.529	0.829	0.933
			WN18RR		
Uncal	4912	0.439	0.394	0.453	0.532
PS	4909 ↑	0.439	0.394	0.453	0.532
HB	19006 ↓	0.100 ↓	0.090 ↑	0.108 ↓	0.117 ↓
IR	18174 ↓	0.099 ↓	0.083 ↑	0.109 ↓	0.124 ↓
BBQ	18192 ↓	0.088 ↓	0.073 ↑	0.100 ↓	0.109 ↓
VS	4888 ↑	0.439	0.394	0.453	0.532
MS	18172 ↓	0.009 ↓	0.003 ↓	0.009 ↓	0.020 ↓
TS	4909 ↑	0.439	0.394	0.453	0.532
Meta-Cal	6157 ↓	0.406 ↓	0.366 ↓	0.419 ↓	0.493 ↓
PTS	4909 ↑	0.439	0.394	0.453	0.532
KGEC	4909 ↑	0.439	0.394	0.453	0.532
			FB15K		
Uncal	41	0.768	0.701	0.813	0.884
PS	41	0.768	0.701	0.813	0.884
HB	1528 ↓	0.630 ↓	0.562 ↓	0.679 ↓	0.748 ↓
IR	952 ↓	0.667 ↓	0.599 ↓	0.713 ↓	0.787 ↓
BBQ	692 ↓	0.603 ↓	0.512 ↓	0.659 ↓	0.775 ↓
VS	39 ↑	0.768	0.701	0.814 ↑	0.885 ↑
MS	3693 ↓	0.025 ↓	0.010 ↓	0.024 ↓	0.055 ↓
TS	41	0.768	0.701	0.813	0.884
Meta-Cal	202 ↓	0.746 ↓	0.680 ↓	0.790 ↓	0.861 ↓
PTS	41	0.768	0.701	0.813	0.884
KGEC	41	0.768	0.701	0.813	0.884
			FB15K-237		
Uncal	174	0.309	0.222	0.337	0.484
PS	174	0.309	0.222	0.337	0.484
HB	2695 ↓	0.256 ↓	0.184 ↓	0.286 ↓	0.401 ↓
IR	2156 ↓	0.280 ↓	0.205 ↑	0.311 ↓	0.427 ↓
BBQ	1562 ↓	0.235 ↓	0.163 ↓	0.259 ↓	0.378 ↓
VS	172 ↑	0.305 ↓	0.216 ↓	0.333 ↓	0.484
MS	3704 ↓	0.033 ↓	0.014 ↓	0.032 ↓	0.070 ↓
TS	174	0.309	0.222	0.337	0.484
Meta-Cal	259 ↓	0.300 ↓	0.213 ↓	0.327 ↓	0.474 ↓
PTS	5659 ↓	0.222 ↓	0.222	0.222 ↓	0.223 ↓
KGEC	174	0.309	0.222	0.337	0.484

model, as shown in Figure 3 and Figure 4. These examples highlight how calibrated probabilities offer more interpretable and informative confidence scores compared to raw, uncalibrated scores.

Case 1: (Greece, _member_of_domain_region, ?) The ground-truth answer for this query is *sibyl*, which is ranked third among the candidate entities based on the model's raw scores. However, the uncalibrated scores do not reflect a meaningful confidence distribution, with the top-ranked entity *Greece* receiving a score of -0.1873 and the correct answer *sibyl* receiving -0.5992, a difference that is difficult to interpret probabilistically.

After applying KGEC calibration, the corresponding probabilities become more interpretable:

- *Greece*: 0.0302
- *Holy See*: 0.0272
- *sibyl* (true answer): 0.0200

These calibrated probabilities clearly reflect the uncertainty inherent in the model's prediction. Although the correct answer is not ranked first, its probability is close to that of the top candidates, suggesting it is still a plausible prediction. This shows that KGEC can better express confidence levels, especially in cases with closely competing candidates.

Case 2: (North_Atlantic_Treaty_Organization, _member_meronym, ?) In this case, the true answer is *Netherlands*, which is correctly ranked second. The raw score of the correct answer (1.6756) is only slightly lower than that of the top-ranked entity *North Atlantic Treaty Organization* (1.9763),

Table 10: Effect of different calibration methods on the performance of the RotatE model across various datasets.

Method	*MR*	*MRR*	*HITS@1*	*HITS@3*	*HITS@10*
			WN18		
Uncal	270	0.950	0.944	0.952	0.960
PS	270	0.950	0.944	0.952	0.960
HB	13910 ↓	0.279 ↓	0.263 ↓	0.294 ↓	0.299 ↓
IR	13962 ↓	0.297 ↓	0.286 ↓	0.308 ↓	0.313 ↓
BBQ	13801 ↓	0.271 ↓	0.253 ↓	0.286 ↓	0.297 ↓
VS	270	0.950	0.944	0.952	0.960
MS	16626 ↓	0.013 ↓	0.005 ↓	0.013 ↓	0.027 ↓
TS	270	0.950	0.944	0.952	0.960
Meta-Cal	1917 ↓	0.905 ↓	0.904 ↓	0.905 ↓	0.905 ↓
PTS	474 ↓	0.949 ↓	0.944	0.951 ↓	0.958 ↓
KGEC	270	0.950	0.944	0.952	0.960
			WN18RR		
Uncal	3421	0.476	0.429	0.496	0.570
PS	3421	0.476	0.429	0.497 ↑	0.570
HB	18719 ↓	0.114 ↓	0.104 ↓	0.122 ↓	0.127 ↓
IR	18047 ↓	0.118 ↓	0.103 ↓	0.128 ↓	0.143 ↓
BBQ	18189 ↓	0.086 ↓	0.073 ↓	0.095 ↓	0.105 ↓
VS	3422 ↓	0.476	0.429	0.497 ↑	0.570
MS	18195 ↓	0.009 ↓	0.003 ↓	0.008 ↓	0.020 ↓
TS	3421	0.476	0.429	0.497 ↑	0.570
Meta-Cal	6168 ↓	0.448 ↓	0.409 ↓	0.464 ↓	0.523 ↓
PTS	3776 ↓	0.474 ↓	0.429	0.493 ↓	0.564 ↓
KGEC	3421	0.476	0.429	0.497 ↑	0.570
			FB15K		
Uncal	41	0.791	0.739	0.825	0.881
PS	41	0.791	0.739	0.825	0.881
HB	1843 ↓	0.642 ↓	0.588 ↓	0.682 ↓	0.731 ↓
IR	961 ↓	0.696 ↓	0.635 ↓	0.741 ↓	0.799 ↓
BBQ	1027 ↓	0.662 ↓	0.599 ↓	0.709 ↓	0.768 ↓
VS	42 ↓	0.791	0.739	0.825	0.880 ↓
MS	3693 ↓	0.025 ↓	0.010 ↓	0.024 ↓	0.055 ↓
TS	41	0.791	0.739	0.825	0.881
Meta-Cal	457 ↓	0.750 ↓	0.700 ↓	0.783 ↓	0.835 ↓
PTS	1122 ↓	0.763 ↓	0.739	0.782 ↓	0.801 ↓
KGEC	41	0.791	0.739	0.825	0.881
			FB15K-237		
Uncal	178	0.336	0.239	0.374	0.530
PS	178	0.336	0.239	0.374	0.530
HB	3458 ↓	0.285 ↓	0.221 ↓	0.317 ↓	0.412 ↓
IR	2131 ↓	0.307 ↓	0.232 ↓	0.340 ↓	0.455 ↓
BBQ	2292 ↓	0.275 ↓	0.204 ↓	0.305 ↓	0.415 ↓
VS	179 ↓	0.336	0.239	0.374	0.530
MS	3704 ↓	0.033 ↓	0.014 ↓	0.032 ↓	0.070 ↓
TS	178	0.336	0.239	0.374	0.530
Meta-Cal	246 ↓	0.328 ↓	0.232 ↓	0.365 ↓	0.522 ↓
PTS	179 ↓	0.336	0.239	0.374	0.530
KGEC	178	0.336	0.239	0.374	0.530

but the significance of this difference is unclear without proper calibration.

With KGEC, the calibrated probabilities provide a more informative picture:

- *North Atlantic Treaty Organization*: 0.3756
- *Netherlands* (true answer): 0.2781
- *European Union*: 0.1382

Here, although the true answer is not ranked first, its calibrated probability is still relatively high, reflecting the model's uncertainty and partially shared semantics among top candidates. This enables downstream applications to interpret and potentially leverage multiple candidates rather than over-committing to the top-1 prediction.

Query: ('North_Atlantic_Treaty_Organization, _member_meronym, ?)

True answer: Netherlands

Ranked candidate entities	Uncalibrated scores	Calibrated probabilities
'North_Atlantic_Treaty_Organization	1.9763	0.3756
Netherlands	1.6756	0.2781
European_Union	0.9763	0.1382
Benelux	0.9763	0.1382
Apeldoorn	-0.4998	0.0316
Leiden	-0.5236	0.0308
Frisian_Islands	-0.5844	0.0290
Friesland	-0.6578	0.0270
Netherlander	-0.6780	0.0264
British_Commonwealth	-0.7083	0.0256

Figure 4: Case 2 from the WN18RR dataset using the TransE model.

Insights. These case studies demonstrate that:

- KGEC enhances the interpretability of model outputs by transforming unnormalized scores into well-calibrated probabilities.
- It allows more accurate reflection of confidence levels, particularly in ambiguous or competitive ranking situations.
- Even when the top-1 prediction is incorrect, KGEC highlights alternative candidates with meaningful confidence, which is valuable for applications such as knowledge graph reasoning, question answering, and downstream ensemble methods.

Overall, these cases exemplify the effectiveness of KGEC in improving the trustworthiness and usability of KGE models.

Table 11: Effect of each component in KGEC on the performance and efficiency of various KGE models across multiple datasets. For all the five metrics, the lower the better.

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*	
KGEC-loss-MBS-JSS	0.642	0.195	0.637	0.213	0.852	0.423	0.691	0.228	0.528	0.389	0.689	0.220	0.805	0.383	0.671	0.222	0.487
KGEC-loss-MBS	0.634	0.196	0.637	0.213	0.852	0.423	0.691	0.228	0.528	0.389	0.688	0.220	0.821	0.383	0.672	0.222	0.487
KGEC-loss	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	0.459	0.150	0.833	0.418	0.678	0.189	0.446	0.383	0.683	0.178	0.467	0.307	0.466	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*	
KGEC-loss-MBS-JSS	0.517	0.285	0.636	0.168	0.852	0.423	0.691	0.227	0.527	0.389	0.688	0.220	0.405	0.383	0.636	0.220	0.454
KGEC-loss-MBS	0.516	0.285	0.630	0.168	0.852	0.423	0.690	0.227	0.527	0.389	0.688	0.220	0.402	0.383	0.636	0.220	0.454
KGEC-loss	0.510	0.283	7.651	0.943	0.823	0.350	0.670	0.161	0.501	0.388	0.666	0.163	0.401	0.278	3.092	0.308	1.074
KGEC	0.131	0.277	0.293	0.082	0.833	0.418	0.465	0.207	0.457	0.383	0.516	0.199	0.467	0.306	0.466	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*	
KGEC-loss-MBS-JSS	2.827	6.544	3.270	5.177	6.830	7.777	5.329	7.294	7.384	7.820	5.294	7.485	1.313	6.107	3.465	5.531	5.590
KGEC-loss-MBS	2.834	6.544	3.310	5.189	6.831	7.778	5.311	7.300	7.384	7.812	5.265	7.479	1.304	6.107	3.470	5.521	5.590
KGEC-loss	2.834	6.330	0.687	4.093	4.856	7.636	6.732	3.811	5.407	7.772	6.444	3.950	1.309	6.327	5.014	6.156	4.960
KGEC	2.462	5.965	2.536	2.889	4.350	6.965	1.357	2.911	2.843	7.119	1.319	3.106	1.036	4.698	2.033	2.743	3.396

Training Time / s	***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*	
KGEC-loss-MBS-JSS	39.769	24.194	139.544	54.700	40.996	23.856	148.151	50.186	39.602	24.557	147.145	52.021	39.659	24.270	153.269	52.023	65.871
KGEC-loss-MBS	2.894	1.638	11.442	3.598	2.714	1.611	10.166	3.546	2.661	1.645	10.147	3.603	2.825	1.608	10.760	3.695	4.659
KGEC-loss	2.785	1.676	10.305	3.598	2.915	1.650	10.246	3.597	2.660	1.644	10.490	3.527	2.671	1.605	10.747	3.578	4.606
KGEC	2.727	1.776	10.873	3.602	2.698	1.727	10.560	3.624	2.741	1.696	10.645	3.705	2.662	1.658	10.758	4.003	4.716

Memory Usage / MB	***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*	
KGEC-loss-MBS-JSS	161.801	126.141	58.465	50.965	170.859	111.168	41.953	65.574	174.496	124.086	56.414	56.473	160.766	94.270	45.000	63.301	97.608
KGEC-loss-MBS	29.027	27.121	6.871	17.969	31.258	26.676	7.535	10.391	25.426	27.184	8.750	14.277	32.906	30.742	6.422	17.961	20.032
KGEC-loss	29.414	27.145	6.898	18.016	25.645	26.879	8.145	10.375	32.254	26.613	8.695	14.320	32.754	30.965	10.172	17.316	20.350
KGEC	30.484	28.289	7.570	15.273	26.652	32.176	9.535	15.285	34.316	32.047	10.531	13.492	34.320	32.191	7.551	16.930	**21.665**

Table 12: Effect of different number of bins in KGEC on the performance of various KGE models across multiple datasets. For all the three metrics, the lower the better.

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*	
Bin=1	0.702	0.196	0.586	0.198	0.851	0.422	0.642	0.227	0.527	0.389	0.687	0.221	0.904	0.382	0.663	0.222	0.489
Bin=2	0.305	0.316	0.581	0.184	0.850	0.422	0.677	0.190	0.521	0.387	0.683	0.219	0.476	0.305	0.671	0.105	0.431
Bin=3	0.214	0.238	0.498	0.183	0.848	0.422	0.677	0.190	0.515	0.385	0.683	0.180	0.467	0.293	0.653	0.101	0.409
Bin=4	0.245	0.249	0.491	0.180	0.848	0.421	0.677	0.190	0.447	0.385	0.682	0.179	0.486	0.286	0.646	0.098	0.407
Bin=5	0.235	0.262	0.479	0.182	0.848	0.420	0.677	0.189	0.447	0.385	0.682	0.179	0.470	0.297	0.622	0.102	0.405
Bin=6	0.211	0.260	0.514	0.170	0.848	0.419	0.677	0.189	0.447	0.384	0.682	0.179	0.487	0.290	0.584	0.112	0.403
Bin=7	0.159	0.273	0.457	0.147	0.848	0.418	0.678	0.189	0.447	0.384	0.682	0.179	0.451	0.304	0.593	0.104	0.394
Bin=8	0.194	0.269	0.460	0.161	0.848	0.418	0.678	0.189	0.446	0.384	0.682	0.179	0.464	0.307	0.529	0.126	0.396
Bin=9	0.181	0.276	0.444	0.160	0.841	0.418	0.678	0.189	0.446	0.384	0.683	0.178	0.464	0.305	0.498	0.157	0.394
Bin=10	0.171	0.280	0.459	0.150	0.833	0.418	0.678	0.189	0.446	0.383	0.683	0.178	0.467	0.307	0.466	0.094	0.388
Bin=11	0.164	0.283	0.416	0.137	0.833	0.418	0.678	0.189	0.446	0.383	0.683	0.178	0.476	0.316	0.491	0.093	0.387
Bin=12	0.163	0.281	0.388	0.162	0.833	0.418	0.678	0.189	0.446	0.383	0.683	0.178	0.475	0.316	0.475	0.100	0.386
Bin=13	0.148	0.287	0.370	0.123	0.835	0.418	0.678	0.189	0.446	0.383	0.683	0.178	0.471	0.317	0.459	0.086	0.379
Bin=14	0.125	0.293	0.376	0.140	0.835	0.417	0.677	0.189	0.446	0.382	0.683	0.178	0.472	0.319	0.458	0.116	0.381
Bin=15	0.102	0.294	0.336	0.129	0.824	0.417	0.678	0.189	0.446	0.381	0.682	0.178	0.475	0.318	0.461	0.076	0.374
Bin=16	0.154	0.296	0.349	0.061	0.769	0.416	0.677	0.189	0.446	0.379	0.682	0.178	0.472	0.319	0.494	0.087	0.373
Bin=17	0.120	0.296	0.313	0.064	0.764	0.415	0.585	0.189	0.446	0.377	0.682	0.178	0.478	0.324	0.490	0.091	0.363
Bin=18	0.115	0.293	0.256	0.085	0.749	0.415	0.589	0.189	0.446	0.377	0.640	0.178	0.472	0.325	0.489	0.134	0.360
Bin=19	0.113	0.293	0.256	0.084	0.749	0.415	0.579	0.189	0.446	0.377	0.603	0.178	0.482	0.326	0.488	0.062	0.352
Bin=20	0.135	0.298	0.261	0.073	0.753	0.416	0.580	0.189	0.445	0.376	0.577	0.178	0.478	0.319	0.493	0.046	0.351

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*	
Bin=1	0.598	0.285	0.565	0.158	0.851	0.422	0.633	0.227	0.527	0.389	0.686	0.221	0.385	0.382	0.602	0.220	0.447
Bin=2	0.318	0.323	0.406	0.128	0.849	0.422	0.494	0.217	0.519	0.387	0.538	0.217	0.476	0.242	0.495	0.093	0.383
Bin=3	0.232	0.243	0.378	0.104	0.848	0.422	0.491	0.217	0.511	0.385	0.533	0.217	0.467	0.226	0.471	0.076	0.364
Bin=4	0.171	0.253	0.328	0.097	0.848	0.421	0.488	0.213	0.460	0.385	0.530	0.213	0.449	0.281	0.450	0.082	0.354
Bin=5	0.155	0.264	0.307	0.099	0.848	0.420	0.484	0.211	0.460	0.385	0.529	0.210	0.458	0.297	0.441	0.069	0.352
Bin=6	0.125	0.262	0.261	0.083	0.848	0.419	0.481	0.210	0.460	0.384	0.527	0.204	0.457	0.290	0.454	0.070	0.346
Bin=7	0.135	0.271	0.277	0.061	0.848	0.418	0.478	0.209	0.460	0.384	0.526	0.204	0.451	0.304	0.419	0.067	0.345
Bin=8	0.129	0.267	0.264	0.080	0.848	0.418	0.475	0.208	0.457	0.384	0.522	0.203	0.464	0.307	0.456	0.072	0.347
Bin=9	0.142	0.274	0.270	0.085	0.841	0.418	0.471	0.208	0.457	0.384	0.521	0.200	0.463	0.305	0.454	0.099	0.349
Bin=10	0.131	0.277	0.293	0.082	0.833	0.418	0.465	0.207	0.457	0.383	0.516	0.199	0.467	0.306	0.466	0.063	0.348
Bin=11	0.111	0.280	0.278	0.076	0.833	0.418	0.460	0.207	0.457	0.383	0.513	0.199	0.461	0.316	0.491	0.061	0.347
Bin=12	0.107	0.278	0.259	0.100	0.833	0.418	0.456	0.207	0.457	0.383	0.508	0.199	0.475	0.316	0.475	0.061	0.346
Bin=13	0.128	0.284	0.240	0.077	0.834	0.418	0.450	0.206	0.457	0.383	0.503	0.197	0.471	0.316	0.459	0.059	0.343
Bin=14	0.113	0.291	0.238	0.088	0.834	0.417	0.446	0.205	0.455	0.382	0.497	0.196	0.471	0.319	0.458	0.062	0.342
Bin=15	0.107	0.292	0.239	0.086	0.823	0.417	0.441	0.204	0.455	0.381	0.492	0.195	0.475	0.318	0.461	0.057	0.340
Bin=16	0.111	0.294	0.234	0.063	0.767	0.416	0.436	0.204	0.455	0.379	0.486	0.194	0.472	0.319	0.494	0.053	0.336
Bin=17	0.100	0.296	0.237	0.064	0.762	0.415	0.550	0.204	0.453	0.377	0.481	0.193	0.478	0.324	0.490	0.053	0.342
Bin=18	0.119	0.293	0.248	0.083	0.746	0.415	0.576	0.204	0.453	0.377	0.466	0.192	0.471	0.325	0.489	0.075	0.346
Bin=19	0.115	0.293	0.247	0.090	0.746	0.415	0.534	0.203	0.453	0.377	0.482	0.192	0.482	0.326	0.488	0.046	0.343
Bin=20	0.107	0.298	0.249	0.073	0.750	0.416	0.557	0.203	0.450	0.376	0.553	0.192	0.478	0.319	0.493	0.048	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*	
Bin=1	2.544	6.543	3.910	4.774	6.524	7.496	3.589	7.208	6.944	7.947	5.152	8.395	1.165	6.078	3.184	5.515	5.436
Bin=2	2.865	6.212	3.076	3.365	5.963	7.513	1.350	2.908	5.477	7.578	1.314	6.650	1.281	4.606	1.989	3.056	4.075
Bin=3	2.712	6.265	2.986	3.290	5.633	7.513	1.350	2.908	4.913	7.338	1.315	3.101	1.184	4.472	2.042	2.774	3.737
Bin=4	2.605	6.185	2.845	3.191	5.633	7.299	1.351	2.909	2.843	7.338	1.315	3.101	1.100	4.745	1.974	2.911	3.584
Bin=5	2.537	6.109	2.766	3.061	5.633	7.202	1.351	2.909	2.843	7.338	1.315	3.102	1.088	4.821	2.011	2.713	3.550
Bin=6	2.523	6.085	2.642	3.062	5.633	7.123	1.352	2.909	2.843	7.200	1.316	3.104	1.062	4.716	2.083	2.643	3.518
Bin=7	2.515	6.023	2.641	3.112	5.633	7.009	1.352	2.910	2.843	7.200	1.316	3.104	1.038	4.787	1.939	2.672	3.506
Bin=8	2.499	6.024	2.607	2.956	5.656	7.001	1.353	2.910	2.844	7.200	1.317	3.104	1.051	4.772	2.039	2.580	3.495
Bin=9	2.493	5.987	2.602	2.895	4.835	6.965	1.355	2.910	2.843	7.200	1.317	3.105	1.035	4.711	2.011	2.473	3.421
Bin=10	2.462	5.965	2.536	2.889	4.350	6.965	1.357	2.911	2.843	7.119	1.319	3.106	1.036	4.698	2.033	2.743	3.396
Bin=11	2.466	5.944	2.532	2.898	4.350	6.965	1.358	2.911	2.843	7.119	1.319	3.106	1.013	4.754	2.123	2.743	3.403
Bin=12	2.448	5.943	2.529	2.798	4.350	6.929	1.359	2.911	2.843	7.114	1.321	3.106	1.043	4.731	2.037	2.666	3.383
Bin=13	2.447	5.919	2.486	2.871	4.415	6.933	1.360	2.911	2.843	7.114	1.323	3.107	1.022	4.714	1.959	2.772	3.387
Bin=14	2.437	5.889	2.482	2.809	4.415	6.867	1.363	2.912	2.845	7.027	1.326	3.108	1.017	4.714	1.941	2.593	3.359
Bin=15	2.438	5.880	2.498	2.803	3.987	6.867	1.363	2.912	2.845	6.935	1.329	3.108	1.024	4.689	1.942	2.813	3.340
Bin=16	2.449	5.870	2.463	2.883	2.931	6.773	1.366	2.912	2.845	6.813	1.331	3.109	1.014	4.684	2.060	2.669	3.261
Bin=17	2.434	5.859	2.487	2.855	2.873	6.726	2.678	2.913	2.846	6.722	1.333	3.109	1.023	4.722	2.030	2.638	3.328
Bin=18	2.449	5.865	2.513	2.779	2.711	6.724	2.860	2.913	2.846	6.722	1.860	3.110	1.010	4.715	2.012	2.509	3.350
Bin=19	2.445	5.862	2.498	2.745	2.711	6.733	2.569	2.913	2.846	6.682	2.235	3.110	1.029	4.706	1.996	2.702	3.361
Bin=20	2.438	5.842	2.489	2.791	2.749	6.734	2.710	2.913	2.849	6.635	2.743	3.110	1.014	4.611	2.010	2.818	3.404

Table 13: Effect of different initial temperature parameters in KGEC on the performance of various KGE models across multiple datasets. For all the three metrics, the lower the better.

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	
init=0	0.382	0.582	0.405	0.213	0.699	0.374	0.677	0.190	0.447	0.349	0.683	0.180	0.118	0.322	0.612	0.237	0.404
init=0.1	0.382	0.582	0.312	0.213	0.699	0.374	0.677	0.190	0.447	0.349	0.683	0.180	0.118	0.389	0.670	0.236	0.406
init=0.2	0.337	0.014	0.626	0.221	0.699	0.374	0.678	0.189	0.447	0.349	0.683	0.179	0.603	0.420	0.719	0.239	0.424
init=0.3	0.696	0.013	0.626	0.221	0.699	0.374	0.678	0.189	0.447	0.349	0.683	0.179	0.939	0.428	0.719	0.239	0.467
init=0.4	0.705	0.014	0.627	0.221	0.699	0.374	0.678	0.189	0.447	0.349	0.683	0.178	0.944	0.426	0.729	0.239	0.469
init=0.5	0.706	0.014	0.645	0.221	0.699	0.279	0.678	0.189	0.447	0.222	0.683	0.178	0.944	0.397	0.709	0.239	0.453
init=0.6	0.706	0.268	0.624	0.233	0.699	0.348	0.678	0.189	0.447	0.336	0.683	0.178	0.944	0.320	0.668	0.239	0.472
init=0.7	0.706	0.390	0.566	0.233	0.699	0.384	0.678	0.189	0.447	0.356	0.683	0.178	0.907	0.244	0.597	0.239	0.468
init=0.8	0.706	0.424	0.520	0.226	0.793	0.404	0.678	0.189	0.446	0.370	0.683	0.178	0.355	0.215	0.530	0.173	0.431
init=0.9	0.444	0.342	0.496	0.200	0.823	0.412	0.678	0.189	0.446	0.376	0.683	0.178	0.402	0.273	0.492	0.124	0.410
init=1.0	0.171	0.280	0.459	0.150	0.833	0.418	0.678	0.189	0.446	0.383	0.683	0.178	0.467	0.307	0.466	0.094	0.388
init=1.1	0.199	0.232	0.419	0.109	0.848	0.420	0.678	0.189	0.446	0.386	0.683	0.178	0.547	0.331	0.495	0.098	0.391
init=1.2	0.278	0.195	0.365	0.073	0.849	0.422	0.606	0.189	0.485	0.388	0.683	0.178	0.608	0.350	0.528	0.101	0.394
init=1.3	0.340	0.165	0.335	0.051	0.850	0.423	0.592	0.189	0.513	0.388	0.683	0.178	0.656	0.362	0.551	0.106	0.399
init=1.4	0.390	0.141	0.360	0.064	0.851	0.424	0.618	0.189	0.519	0.390	0.683	0.178	0.696	0.373	0.568	0.108	0.409
init=1.5	0.430	0.122	0.388	0.082	0.852	0.425	0.599	0.190	0.524	0.390	0.602	0.178	0.727	0.382	0.581	0.133	0.413
init=1.6	0.463	0.105	0.411	0.097	0.852	0.425	0.625	0.209	0.525	0.391	0.597	0.178	0.752	0.390	0.594	0.128	0.421
init=1.7	0.491	0.090	0.429	0.110	0.852	0.425	0.611	0.213	0.526	0.391	0.609	0.193	0.773	0.395	0.609	0.150	0.429
init=1.8	0.515	0.079	0.447	0.121	0.853	0.426	0.607	0.223	0.527	0.392	0.611	0.186	0.791	0.400	0.622	0.162	0.435
init=1.9	0.534	0.069	0.462	0.133	0.853	0.426	0.616	0.223	0.527	0.392	0.614	0.198	0.807	0.404	0.631	0.167	0.441
init=2.0	0.550	0.061	0.475	0.141	0.853	0.426	0.644	0.225	0.528	0.392	0.613	0.209	0.820	0.407	0.638	0.172	0.447

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	
init=0	55.708	27.896	26.189	4.896	0.679	0.163	0.515	0.226	0.471	0.240	0.547	0.228	50.620	3.966	4.828	1.306	11.155
init=0.1	37.388	18.451	12.026	4.896	0.679	0.163	0.515	0.226	0.471	0.240	0.547	0.228	27.979	3.158	4.275	1.278	7.032
init=0.2	8.518	4.825	2.672	1.805	0.688	0.160	0.467	0.210	0.458	0.238	0.517	0.201	8.266	1.911	2.159	0.995	2.131
init=0.3	4.237	2.449	2.671	1.805	0.688	0.160	0.466	0.208	0.458	0.238	0.517	0.200	3.453	1.006	2.154	0.995	1.357
init=0.4	2.585	1.508	2.646	1.814	0.688	0.160	0.465	0.208	0.457	0.238	0.516	0.200	1.404	0.483	0.288	0.995	0.916
init=0.5	1.535	0.994	0.311	1.795	0.688	0.265	0.465	0.207	0.457	0.208	0.516	0.200	0.726	0.195	0.298	0.995	0.616
init=0.6	0.947	0.721	0.249	0.392	0.688	0.347	0.465	0.207	0.457	0.335	0.516	0.200	0.314	0.119	0.327	0.994	0.455
init=0.7	0.573	0.542	0.242	0.290	0.689	0.384	0.465	0.207	0.457	0.355	0.516	0.199	0.105	0.152	0.362	0.994	0.408
init=0.8	0.307	0.422	0.257	0.209	0.792	0.404	0.465	0.207	0.457	0.369	0.516	0.199	0.230	0.213	0.402	0.113	0.348
init=0.9	0.144	0.339	0.275	0.141	0.822	0.412	0.465	0.207	0.457	0.376	0.516	0.199	0.366	0.273	0.427	0.070	0.343
init=1.0	0.131	0.277	0.293	0.082	0.833	0.418	0.465	0.207	0.457	0.383	0.516	0.199	0.467	0.306	0.466	0.063	0.348
init=1.1	0.200	0.230	0.309	0.050	0.847	0.420	0.465	0.207	0.457	0.386	0.516	0.199	0.547	0.331	0.495	0.064	0.358
init=1.2	0.278	0.195	0.331	0.042	0.848	0.422	0.582	0.207	0.478	0.388	0.516	0.199	0.608	0.350	0.528	0.088	0.379
init=1.3	0.340	0.167	0.359	0.048	0.850	0.423	0.493	0.207	0.509	0.388	0.516	0.199	0.656	0.362	0.551	0.094	0.385
init=1.4	0.390	0.145	0.385	0.059	0.851	0.424	0.607	0.207	0.516	0.390	0.517	0.199	0.696	0.373	0.568	0.095	0.401
init=1.5	0.430	0.126	0.409	0.074	0.851	0.425	0.563	0.197	0.523	0.390	0.501	0.199	0.727	0.382	0.581	0.122	0.406
init=1.6	0.463	0.110	0.429	0.094	0.852	0.425	0.618	0.209	0.525	0.391	0.509	0.199	0.752	0.390	0.594	0.116	0.417
init=1.7	0.491	0.096	0.445	0.109	0.852	0.425	0.593	0.211	0.526	0.391	0.597	0.199	0.773	0.395	0.609	0.142	0.429
init=1.8	0.515	0.086	0.462	0.121	0.853	0.426	0.584	0.222	0.527	0.392	0.600	0.197	0.791	0.400	0.622	0.158	0.435
init=1.9	0.534	0.076	0.474	0.133	0.853	0.426	0.604	0.222	0.527	0.392	0.606	0.200	0.807	0.404	0.631	0.163	0.441
init=2.0	0.550	0.068	0.485	0.141	0.853	0.426	0.644	0.224	0.528	0.392	0.604	0.203	0.820	0.407	0.638	0.169	0.447

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	
init=0	-1.714	1.977	-0.694	0.791	2.287	3.210	1.349	2.908	2.841	3.253	1.314	3.099	-3.229	1.540	-0.515	1.145	1.223
init=0.1	-1.422	2.230	-0.296	0.791	2.287	3.210	1.349	2.908	2.841	3.253	1.314	3.099	-2.814	1.620	-0.452	1.150	1.317
init=0.2	-0.209	3.154	0.492	1.014	2.287	3.211	1.356	2.909	2.843	3.253	1.318	3.104	-1.858	1.738	-0.316	1.166	1.591
init=0.3	0.378	3.833	0.493	1.014	2.288	3.211	1.356	2.910	2.843	3.253	1.319	3.104	-1.191	2.238	-0.315	1.165	1.744
init=0.4	0.786	4.312	0.498	1.012	2.288	3.211	1.356	2.910	2.843	3.253	1.319	3.104	-0.549	2.683	1.189	1.166	1.961
init=0.5	1.172	4.722	1.926	1.016	2.288	4.151	1.356	2.910	2.843	4.318	1.319	3.105	-0.187	3.085	1.339	1.166	2.283
init=0.6	1.494	5.037	2.077	2.197	2.288	4.852	1.356	2.910	2.843	5.461	1.319	3.105	0.125	3.460	1.501	1.166	2.574
init=0.7	1.777	5.317	2.191	2.374	2.288	5.459	1.356	2.910	2.843	5.889	1.319	3.105	0.395	3.839	1.682	1.167	2.744
init=0.8	2.037	5.560	2.322	2.542	3.267	6.058	1.356	2.910	2.843	6.328	1.319	3.105	0.637	4.125	1.825	2.429	3.042
init=0.9	2.261	5.773	2.434	2.710	3.937	6.483	1.357	2.910	2.843	6.642	1.319	3.106	0.847	4.456	1.902	2.582	3.223
init=1.0	2.462	5.965	2.536	2.889	4.350	6.965	1.357	2.911	2.843	7.119	1.319	3.106	1.036	4.698	2.033	2.743	3.396
init=1.1	2.651	6.142	2.635	3.006	5.567	7.214	1.357	2.911	2.844	7.429	1.319	3.106	1.219	4.929	2.149	2.727	3.575
init=1.2	2.820	6.302	2.743	3.155	5.739	7.427	2.960	2.911	3.803	7.687	1.319	3.106	1.383	5.144	2.293	3.136	3.871
init=1.3	2.975	6.453	2.844	3.328	6.148	7.606	2.410	2.911	4.791	7.751	1.319	3.106	1.538	5.312	2.407	3.201	4.006
init=1.4	3.121	6.597	2.934	3.421	6.262	7.852	3.194	2.911	5.248	8.071	1.319	3.106	1.688	5.502	2.499	3.216	4.184
init=1.5	3.256	6.729	3.043	3.579	6.591	8.080	2.817	3.398	5.977	8.213	2.415	3.106	1.819	5.678	2.578	3.521	4.425
init=1.6	3.380	6.863	3.137	3.715	6.859	8.228	3.322	4.532	6.350	8.329	2.543	3.107	1.942	5.855	2.668	3.466	4.643
init=1.7	3.502	6.995	3.211	3.834	6.977	8.418	3.058	4.880	6.663	8.459	3.156	4.061	2.056	6.014	2.776	3.708	4.861
init=1.8	3.618	7.108	3.298	3.936	7.242	8.485	2.979	6.043	7.090	8.595	3.186	3.850	2.165	6.183	2.880	3.883	5.034
init=1.9	3.724	7.221	3.376	4.053	7.456	8.669	3.164	5.999	7.228	8.794	3.245	4.284	2.274	6.327	2.960	3.945	5.170
init=2.0	3.822	7.331	3.447	4.136	7.635	8.782	3.690	6.353	7.515	8.882	3.230	5.035	2.375	6.462	3.029	4.035	5.360

Table 14: Effect of different learning rate in KGEC on the performance of various KGE models across multiple datasets. For all the three metrics, the lower the better.

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*	
lr=0.001	0.184	0.229	0.469	0.098	0.852	0.423	0.644	0.228	0.528	0.389	0.659	0.220	0.492	0.381	0.498	0.191	0.405
lr=0.002	0.172	0.231	0.517	0.052	0.852	0.423	0.608	0.227	0.527	0.388	0.596	0.218	0.476	0.371	0.456	0.092	0.388
lr=0.003	0.199	0.265	0.534	0.099	0.851	0.423	0.593	0.226	0.527	0.388	0.594	0.212	0.481	0.361	0.467	0.092	0.395
lr=0.004	0.203	0.276	0.535	0.138	0.851	0.422	0.595	0.222	0.526	0.387	0.605	0.191	0.478	0.353	0.480	0.098	0.398
lr=0.005	0.179	0.281	0.521	0.157	0.850	0.422	0.594	0.213	0.525	0.386	0.612	0.178	0.479	0.344	0.476	0.092	0.394
lr=0.006	0.202	0.286	0.520	0.160	0.849	0.421	0.678	0.207	0.522	0.386	0.683	0.178	0.477	0.333	0.483	0.091	0.405
lr=0.007	0.201	0.291	0.497	0.166	0.849	0.421	0.678	0.189	0.517	0.385	0.683	0.178	0.472	0.323	0.462	0.110	0.401
lr=0.008	0.188	0.287	0.491	0.163	0.846	0.419	0.678	0.189	0.495	0.384	0.683	0.178	0.466	0.317	0.455	0.127	0.398
lr=0.009	0.188	0.284	0.478	0.167	0.843	0.419	0.678	0.189	0.446	0.384	0.683	0.178	0.467	0.313	0.463	0.097	0.392
lr=0.010	0.171	0.280	0.459	0.150	0.833	0.418	0.678	0.189	0.446	0.383	0.683	0.178	0.467	0.307	0.466	0.094	0.388
lr=0.020	0.152	0.246	0.380	0.064	0.699	0.392	0.678	0.189	0.446	0.353	0.683	0.178	0.514	0.273	0.560	0.239	0.378
lr=0.030	0.184	0.228	0.460	0.080	0.699	0.374	0.677	0.189	0.446	0.349	0.683	0.178	0.562	0.289	0.606	0.238	0.390
lr=0.040	0.244	0.211	0.507	0.114	0.699	0.374	0.678	0.189	0.446	0.349	0.683	0.178	0.598	0.290	0.638	0.097	0.393
lr=0.050	0.305	0.189	0.530	0.139	0.699	0.374	0.677	0.189	0.446	0.349	0.682	0.178	0.623	0.295	0.652	0.238	0.410
lr=0.060	0.342	0.163	0.554	0.212	0.699	0.364	0.677	0.189	0.446	0.205	0.683	0.178	0.635	0.330	0.674	0.237	0.412
lr=0.070	0.393	0.153	0.569	0.164	0.699	0.374	0.677	0.190	0.526	0.349	0.683	0.178	0.656	0.330	0.679	0.239	0.429
lr=0.080	0.418	0.135	0.577	0.174	0.699	0.407	0.678	0.189	0.446	0.349	0.682	0.178	0.685	0.332	0.680	0.239	0.429
lr=0.090	0.456	0.130	0.584	0.184	0.826	0.374	0.677	0.189	0.446	0.385	0.683	0.178	0.716	0.350	0.693	0.237	0.444
lr=0.100	0.494	0.108	0.590	0.193	0.699	0.374	0.677	0.189	0.527	0.349	0.683	0.178	0.742	0.331	0.696	0.239	0.442

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*	
lr=0.001	0.220	0.237	0.276	0.098	0.852	0.423	0.644	0.227	0.527	0.388	0.659	0.219	0.492	0.380	0.498	0.190	0.396
lr=0.002	0.171	0.236	0.248	0.037	0.851	0.423	0.587	0.226	0.527	0.388	0.562	0.216	0.459	0.370	0.456	0.069	0.364
lr=0.003	0.119	0.263	0.247	0.049	0.851	0.423	0.513	0.225	0.526	0.388	0.547	0.205	0.457	0.361	0.445	0.069	0.356
lr=0.004	0.108	0.273	0.252	0.075	0.851	0.422	0.486	0.220	0.525	0.387	0.499	0.198	0.457	0.352	0.437	0.063	0.350
lr=0.005	0.109	0.278	0.256	0.092	0.850	0.422	0.527	0.211	0.524	0.386	0.497	0.199	0.461	0.344	0.440	0.067	0.354
lr=0.006	0.107	0.283	0.264	0.095	0.849	0.421	0.465	0.208	0.521	0.386	0.516	0.199	0.463	0.333	0.435	0.065	0.351
lr=0.007	0.112	0.288	0.264	0.101	0.849	0.421	0.465	0.207	0.514	0.385	0.516	0.199	0.465	0.323	0.448	0.066	0.351
lr=0.008	0.122	0.284	0.272	0.097	0.845	0.419	0.465	0.207	0.488	0.384	0.516	0.199	0.466	0.317	0.452	0.071	0.350
lr=0.009	0.127	0.281	0.284	0.100	0.842	0.419	0.465	0.207	0.457	0.384	0.516	0.199	0.467	0.313	0.463	0.062	0.349
lr=0.010	0.131	0.277	0.293	0.082	0.833	0.418	0.465	0.207	0.457	0.383	0.516	0.199	0.467	0.306	0.466	0.063	0.348
lr=0.020	0.146	0.244	0.400	0.048	0.689	0.392	0.465	0.207	0.457	0.352	0.517	0.199	0.514	0.273	0.560	0.994	0.404
lr=0.030	0.163	0.227	0.471	0.075	0.689	0.160	0.484	0.207	0.457	0.238	0.517	0.199	0.562	0.289	0.606	1.010	0.397
lr=0.040	0.244	0.211	0.511	0.114	0.689	0.160	0.466	0.207	0.457	0.238	0.522	0.199	0.598	0.290	0.638	0.085	0.352
lr=0.050	0.305	0.191	0.535	0.139	0.689	0.160	0.486	0.207	0.457	0.238	0.529	0.199	0.623	0.295	0.652	1.115	0.426
lr=0.060	0.342	0.167	0.554	2.220	0.689	0.364	0.493	0.207	0.457	0.189	0.519	0.199	0.635	0.330	0.674	1.067	0.569
lr=0.070	0.393	0.157	0.569	0.164	0.689	0.160	0.493	0.212	0.526	0.238	0.518	0.199	0.656	0.330	0.679	0.987	0.436
lr=0.080	0.418	0.139	0.577	0.174	0.689	0.407	0.469	0.207	0.457	0.238	0.533	0.199	0.685	0.332	0.680	0.986	0.449
lr=0.090	0.456	0.136	0.584	0.184	0.826	0.160	0.503	0.208	0.457	0.383	0.517	0.200	0.716	0.350	0.693	1.142	0.470
lr=0.100	0.494	0.114	0.590	0.193	0.689	0.160	0.498	0.208	0.527	0.240	0.521	0.200	0.742	0.331	0.696	0.997	0.450

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*	
lr=0.001	2.751	6.481	2.469	3.771	6.777	7.749	3.691	7.271	7.279	7.797	4.071	7.207	1.254	5.847	2.159	4.410	5.061
lr=0.002	2.505	6.267	2.341	3.233	6.634	7.665	3.006	6.945	7.073	7.729	2.871	6.351	1.014	5.546	1.997	2.862	4.627
lr=0.003	2.471	6.029	2.309	3.034	6.514	7.599	2.511	6.618	6.818	7.646	2.763	5.263	1.008	5.357	1.961	2.860	4.423
lr=0.004	2.468	5.984	2.311	2.924	6.292	7.493	2.301	5.853	6.526	7.562	2.353	3.975	1.010	5.217	1.934	2.703	4.182
lr=0.005	2.473	5.966	2.331	2.865	6.188	7.455	2.587	4.799	6.230	7.495	2.256	3.106	1.019	5.101	1.942	2.817	4.039
lr=0.006	2.464	5.947	2.353	2.853	5.874	7.356	1.357	4.372	5.670	7.383	1.319	3.106	1.024	4.965	1.927	2.793	3.798
lr=0.007	2.462	5.931	2.406	2.834	5.773	7.280	1.357	2.911	5.093	7.362	1.319	3.106	1.028	4.857	1.970	2.635	3.645
lr=0.008	2.466	5.942	2.444	2.845	5.288	7.094	1.357	2.911	4.088	7.253	1.319	3.106	1.032	4.800	1.984	2.573	3.531
lr=0.009	2.465	5.951	2.473	2.831	4.963	7.113	1.357	2.911	2.843	7.203	1.319	3.106	1.033	4.757	2.026	2.711	3.441
lr=0.010	2.462	5.965	2.536	2.889	4.350	6.965	1.357	2.911	2.843	7.119	1.319	3.106	1.036	4.698	2.033	2.743	3.396
lr=0.020	2.519	6.087	3.006	3.237	2.288	5.673	1.356	2.911	2.844	5.811	1.319	3.106	1.146	4.446	2.456	1.167	3.086
lr=0.030	2.579	6.160	3.362	3.546	2.288	3.211	1.351	2.910	2.844	3.254	1.320	3.106	1.274	4.564	2.756	1.167	2.856
lr=0.040	2.747	6.236	3.661	3.855	2.288	3.211	1.356	2.910	2.844	3.254	1.320	3.106	1.370	4.575	3.027	3.067	3.052
lr=0.050	2.896	6.335	3.839	4.096	2.288	3.211	1.352	2.910	2.844	3.254	1.315	3.106	1.436	4.608	3.173	1.159	2.989
lr=0.060	2.990	6.475	4.068	0.967	2.288	5.092	1.350	2.910	2.844	4.147	1.322	3.106	1.474	4.934	3.466	1.167	3.037
lr=0.070	3.149	6.532	4.254	4.399	2.288	3.211	1.350	2.909	6.683	3.253	1.321	3.106	1.531	4.920	3.546	1.177	3.352
lr=0.080	3.225	6.648	4.370	4.564	2.288	6.227	1.361	2.910	2.844	3.253	1.315	3.105	1.640	4.945	3.557	1.177	3.339
lr=0.090	3.366	6.692	4.495	4.759	4.059	3.211	1.350	2.910	2.844	7.289	1.319	3.105	1.767	5.153	3.803	1.171	3.581
lr=0.100	3.533	6.874	4.587	4.993	2.288	3.211	1.350	2.910	7.271	3.253	1.321	3.105	1.887	4.928	3.867	1.174	3.534