# DMutDE: Dual-View Mutual Distillation Framework for Knowledge Graph Embeddings

Ruizhou Liu, Zhe Wu, Yiling Wu, Zongsheng Cao, Qianqian Xu, *Senior Member, IEEE,* Qingming Huang, *Fellow, IEEE,*

***Abstract*—Knowledge graphs (KGs) has caught more and more attention in recent years. Currently, in some practical scenarios, KGE models are expected to reduce their spatial complexity without losing much performance to address the challenges of storage limitations and knowledge reasoning efficiency. To achieve this, existing works use one or more large and high-performance teacher models to improve one lightweight student model's performance via knowledge distillation, thus meeting the requirements of some practical complicated applications. However, in resource-constrained scenarios, obtaining high-performance teacher models is challenging due to high training costs and significant storage requirements. Thus, enhancing the student model's performance without large teacher models is crucial. To address this issue, we propose Dual-View Mutual Distillation Framework for Knowledge Graph Embeddings (DMutDE), a distillation framework leveraging mutual learning for peer-to-peer distillation between two KGE models with different architectures. In KGE models, we notice that the way to modeling relational directed edges determines the model view of KGE model for knowledge graph data. Thus, integrating the model views from two different KGE models by knowledge distillation into student KGE model can improve its generalization, so as to increase its performance. To identify an effective dual-view fusion method, we design two modules in DMutDE framework. Specifically, we designed a novel soft label fusion module for noise filtering and response knowledge transfer. Then, we propose an entity embedding distillation module to distill structural features from each other. Finally, we conduct several comprehensive experiments on the standard open-source benchmarks to demonstrate that our framework achieves the state-of-the-art results.**

***Index Terms*—Knowledge graph, knowledge distillation, knowledge graph embedding.**

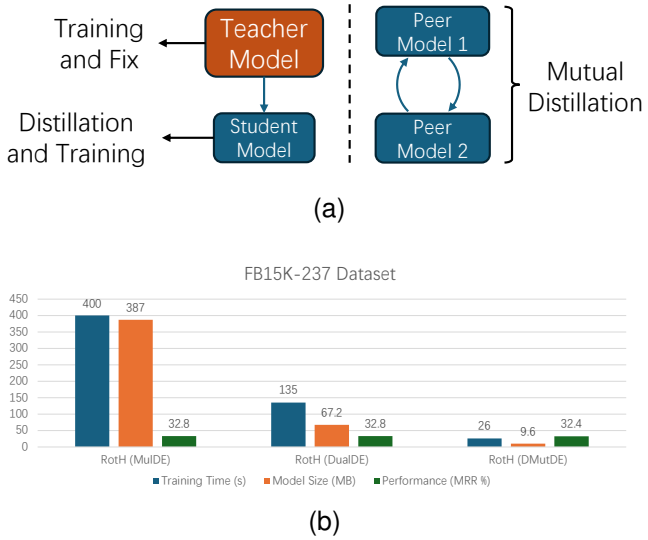


Fig. 1. (a) The left hand side is the pipeline of traditional knowledge distillation framework. The right hand side is the paradigm that our framework used, where the two peer models learn from each other. (b) The comparison of several knowledge distillation frameworks in training time model size and student's performance. The training time includes consuming time of training teacher and student, and the model size is sum of teacher and student.

## I. Introduction

KNOWLEDGE graphs (KGs) can represent the real-world facts in the form of the triplets *i.e.* (head entity, relation entity, tail entity) like YAGO [1], [2], DBPedia [3], and boost the development of amount of semantic applications, *e.g.,* question answering [4]–[6], recommender systems [7], [8], and information retrieval [9], [10]. However, real-world knowledge graphs are typically incomplete, and unlike images where features can be directly extracted using convolutional neural networks, triplet data of KGs cannot be directly processed by downstream models, resulting in low performance on recommander system [11], [12], question answering [13], [14]. To effectively represent relational mappings between entities and fill missing information, many researchers have developed knowledge graph embedding (KGE), which embeds the entities and relational mappings into low-dimensional numerical spaces, portraying the original KG's topological structure and relational mappings. And the representation ability of the KGE methods are further improved from geometric transformation, metric space and linear interaction, such as TransE [14] for translations, RotatE [15], QuatE [16] for rotations, MuRP [17] and AttH [18] for hyperbolic space, DFeildE [19] hyperspherical space and TuckER [20] and ComplEx [21] for linear interaction.

To enhance downstream task performance, currently, applications based on knowledge graphs prefer the KGE method that can learn more accurate representations. A straightforward

Ruizhou Liu is with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100190, China(email:liuruizhou21@mails.ucas.ac.cn); also with the Peng Cheng Laboratory, Shenzhen 518055, China (email:liurzh@pcl.ac.cn).

Zhe Wu and Yiling Wu are with the Peng Cheng Laboratory, Shenzhen 518055, China (email:wuzh02@pcl.ac.cn; wuyl02@pcl.ac.cn).

Zongshneg Cao is with the Institution of Information Engineering, Chinese Academy of Sciences, Beijing 100190, China (email:caozongsheng@iie.ac.cn).

Qianqian Xu is with the Institution of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (email:xuqianqian@ict.ac.cn).

Qingming Huang is with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101408, China (email:qmhuang@ucas.ac.cn).

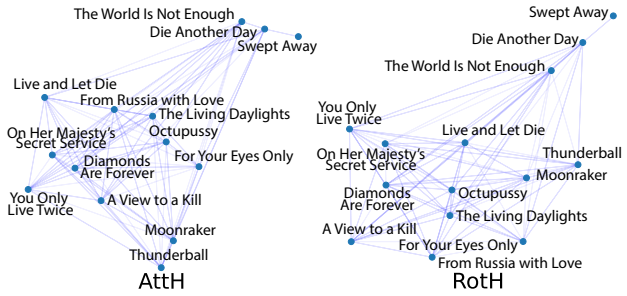


Fig. 2. A visualization of the distributions of partial entity embeddings learned by RotH and AttH on FB15K-237.

yet brute-force approach achieving more accurate representations is increasing the spatial complexity of KGE model, i.e. increasing the embedding dimension of the entities. But this approach only offers limited performance gains to model and significantly increases training time and storage requirements. Thus, some methods use a knowledge distillation (KD) framework [22], where high-performing KGE models serve as teachers to transfer their superior feature representations [23] to a lightweight student model, to achieve high performance in a simpler model without bearing an excessive training burden. However, in resource-constrained scenarios, training a teacher model for a high-performance student model may not be feasible. Thus, the first question should be addressed in this paper is that ***Q1:*** *Can we improve student model performance without training teacher models?*

In addition, we find that the view of modeling KGs data is determined by the way of modeling relation mapping, and the KGE model performances are influenced by different model views. Taking AttH and RotH as examples, on the WN18RR dataset, RotH is better than AttH, while on the FB15K-237 dataset, AttH is better than RotH. This phenomenon arises due to the differences in the way to model relational mappings. Furthermore, this differences leads to the entity embedding distributions learned by these two models are different, as shown in Figure 2. Therefore, we expect to utilize the knowledge distillation to ensemble two distinct views into single KGE model so that improving its generalizations and performance. Therefore, ***Q2:*** *How to integrate the dual views into a single KGE model is worth exploring.*

In this paper, to address the aforementioned two problems, leveraging the peer-to-peer distillation paradigm, we present a novel knowledge distillation framework for improving knowledge graph embedding, termed as **D**ual-View **Mut**ual **D**istillation Framework for KG**E** (DMutDE) employing two different lightweight KGE models (one student model and one auxiliary model) and alternatively distilling with each other. Figure 1a contrasts the traditional distillation paradigm (left) that the student model learns from a pre-trained teacher, with the peer-to-peer distillation paradigm used in our work (right). And the Figure 1b shows the training cost and student model performance of the traditional distillation frameworks (MulDE [24], DualDE [25]) and our framework. Obviously, the baselines suffer from teacher models with higher training time and storage space requirements than ours, while our student model performance is comparable.

Specifically, during distillation, unlike previous works that directly supervise the student with the teacher's predictions, we filter out inference noise from the peer models' predictions before distillation, as the peer models are not well-trained initially. First, we propose a novel soft label fusion module that filters noise by refining the soft labels based on the predictions of student model and auxiliary model for the student model to learn. Second, traditional knowledge distillation methods use distance functions like $l_1$ and $l_2$ to guide the student model in learning the teacher model's feature maps. However, in knowledge graph embedding tasks, the entity embedding distributions of the two models are often too different as the heterogeneity, causing performance degradation when using these traditional methods. To address this, we designed an embedding distillation module that projects the entity embeddings of both models into the learning space via an encoder, enabling the student model to adaptively learn from the teacher model.

We evaluate the performance improvements of the KGE methods distilled by our DMutDE framework on two standard KG benchmarks. Finally, our contributions are summarized as three-fold:

- The current knowledge distillation frameworks for KGE methods necessitate one or more pre-trained large teacher models, posing challenges for the knowledge graphs based applications in some resource-constrained scenarios.
- The way to modeling relational mappings in KGE methods determines the model view for modeling KGs data. A single model view limits and degrades the generalization and performance of KGE models. To address this, we integrate two distinct KGE model views into a student KGE model, enhancing its generalization and performance.
- To address these two issues, we propose DMutDE, a novel knowledge distillation framework. It contains a soft labels fusion mechanism to eliminate noise in the predictions of peer models and an embedding distillation module to reduce the discrepancies of the entity embedding distributions between the student model and auxiliary model. Finally, comprehensive experiments on standard KG benchmarks evaluate the effectiveness and training costs of our framework for link prediction tasks.

The rest of this article is organized as follows. We introduce the related work in Section II. Some preliminaries for the knowledge graph are provided in Section III. Then, our DMutDE framework is described in Section IV. Some related theoretical analysis is described in section V. We report the experimental results and the model analysis in Section VI. Finally, this article is concluded in Section VII. In addition, for a better understanding of the rest of this paper, the main notations/math symbols utilized throughout this article are presented in Table I.

## II. Related Work

### A. Knowledge Graph Embeddings Methods

Knowledge graph embedding is a key technique for learning a representation of knowledge graphs in continuous low-dimensional space to portray topological structure with mul-

TABLE I
NOTATIONS AND EXPLANATIONS

| Notations | Explanations |
|---|---|
| $\mathcal{G}$ | Knowledge graphs |
| $\mathcal{M}$ | Manifold space |
| $\mathcal{E}, \mathcal{R}$ | set of entities, relations |
| $\mathcal{T}, \mathcal{T}'$ | Triplets set, negative triplets set |
| $(h, r, t)$ | head entity, relation type, tail entity |
| $(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$ | Embeddings of head, relation and tail. |
| $\boldsymbol{E}, \boldsymbol{e}$ | Embeddings of all entities and one entity embedding. |
| $\mathbb{E}, \mathbb{H}$ | Euclidean space, Hyperbolic space. |
| $\boldsymbol{r}_{\text{trans}}$ | Relational parameters for translation transformation. |
| $\boldsymbol{r}_{\text{rot}}$ | Relational parameters for rotation transformation. |
| $\boldsymbol{r}_{\text{ref}}$ | Relational parameters for reflective transformation. |
| $\omega_1$ | Weight of local soft-label loss |
| $\omega_2$ | Weight of angle similarity loss |
| $\mu_1$ | Weight of the soft-label loss |
| $\mu_2$ | Weight of the embedding structure loss |
| $\mathcal{P}$ | Score distribution |
| $M_S, M_A$ | Student model and auxiliary model |
| $\boldsymbol{W}, \boldsymbol{b}$ | Learnable weight matrix and bias |
| $\Phi(\cdot)$ | Score function |
| $f_{\text{fusion}}(\cdot, \cdot)$ | The joint soft-labels fusion function |
| $\varphi_{\text{LR}}(\cdot, \cdot)$ | Length ratio function |
| $\varphi_{angle}(\cdot, \cdot)$ | Angle similarity function |
| $g(\cdot)$ | Learning space encoder |
| $\sigma(\cdot)$ | Activate function |
| $\langle \cdot, \cdot \rangle$ | Inner-product |
| $\|\cdot\|_1, \|\cdot\|_2$ | $l_1$-norm, $l_2$-norm |

tiple relation types among entities. There are lots of detailed surveys [26], [27] for knowledge graphs and KGE, here we briefly introduce some typical KGE methods. RESCAL [28] the first proposes to learn multi-relational graphs with tensor decomposition. While DistMult [29] simplifies the relation matrix to diagonal. ComplEx [21] embeds the entities and relations in complex space. TuckER [20] propose a linear model based on Tucker-decomposition for binary adjacent matrix.

TransE [14] is the first method that models relations with geometric transformations for learning knowledge graph representations. TransH [30] solving many-to-many problem, TransR [31] separating relation space and entity space, TransD [32] reducing parameter volume of TransR, etc. RotatE [15] models relation as a rotation transformation on complex plane for solving symmetric and asymmetric relations, QuatE [16] utilizes Quaternions numbers to embed relations in 4d rotations. BiQUE [33] extend relation embeddings with biquaternions, Rotate3D [34] for 3D rotations. While OTE [35] generalizes the rotation transformation in high dimensional for relation embedding. In addition, some studies recognize that geometric transformations can be generalized with algebraic groups. TorusE [36] embeds entities into the compact Lie group, GrpKG [37] summarizes and induces existing KGE methods in algebraic groupoid perspective.

The topological structures of the KGs are essential parts, thus some studies consider learning the topological structures with manifolds. ManifoldE [38] firstly introduces a point-wise manifold for embedding, it relaxes the real-valued point-wise space into manifold space with a more expressive representation from the geometric perspective. Benefiting from measurement characteristics of manifold, several works [39]–[41] has proven that *Riemannian* geometry manifold is equipped with more expressiveness on embedding non-Euclidean data than Euclidean flatten space. MuRP [17] firstly embeds KG in hyperbolic space for portraying hierarchical data. Instead of fixing curvature value, RotH [18] captures hierarchical features and logical patterns by learning curvatures, while AttH [18] possesses the capabilities of modeling more complicated graphs depending on attention mechanism. Moving beyond *Cartesian* coordinate system, HEB [42] uses the polar coordinate system to represent the Poincaré disk for modeling hierarchical structures. HyperKA [43] is the first Graph Neural Network incorporating hyperbolic space for KGE. [44] learns embeddings in a product manifold combining multiple spaces (spherical, hyperbolic, and Euclidean) and introduces a heuristic to estimate the sectional curvature of graph data. While UltraE [45] extends embedding space from hyperbolic to ultra-hyperbolic space [46]. To prompt the expressiveness of manifold, M$^2$GNN utilizes mix-curvature manifold combining with GNN to model topological structures, while GIE [47] proposed multi-manifolds geometry interaction for captures different type structures with different curvatures adaptively. FieldE [48] is no longer limited to constant-curvature manifold, instead constructing vector field based on Ordinary Differential Equation (ODE) for each relation.

Furthermore, several recent works rely on utilizing deep neural networks to explore more complicated semantic interactions between entities and relations. ConvE [49] firstly employs convolutions operations in KGE, while ConvKB [50] keeps the transitional characteristic and shows better experimental performance. While HRAN [51] employs the attention operation in the neural network and GGAE [52] models KGs with attention in a global view. And several works leverage the capabilities of Graph Neural Network (GNN) to capture graph features, like R-GCN [53], KGAT [54]. And AutoSF [55] embeds relations as matrices and recognizes the patterns of parameter distribution on the matrices is important, thus using AutoML [56] techniques to search more expressive patterns. To capture the complex graph information and construct a knowledge-semantic fusion of multiple features, Jiang et al. [57] propose a multisource hierarchical neural network for knowledge graph embedding, which combines from low- to high-dimensional multiple mapping sources, thus facilitating the integration of complex heterogeneous entities and relations. 2. Le et al. [58] believe the rotation in 3D space limits the performance of KGE like Rotate3D [34], and leverage the quaternions algebra group to develop a new 4D rotation transformation, thus obtaining more accurate KG embeddings. 3. Lee et al. [59] notice that general methods in inductive knowledge graph completion assume that all entities can be new, and they do not allow new relations to appear at inference time, which leads to the current methods cannot handle the

new entities accompanying new relations. Therefore, they propose an INGRAM model to address this issue. 4. Xu et al. [60] leverage the pretrained Large-Language-Models (LLMs) to encode the semantics of entities and relation in KG, so as to learning entity and relation representations. However, due to the quality of text and the incomplete structure of KG, they propose MPIKGC, which expands the entity description using the knowledge captured by LLMs.

### B. Knowledge Distillations on KGE Methods

The knowledge distillation framework on KGE methods mainly explore how to use the knowledge distillation techniques [22], [61]–[63] on the existing KGE methods, thus improving the performance of lightweight KGE methods. The goal of Knowledge Distillation (KD) [22] is to transfer the effective knowledge, such as feature representations, from one or more teacher models to a lightweight student model for model compression and faster inference. Large high-performance KGE methods generally suffer from high training costs, e.g. long training time, large energy consumption, and storage. A few studies attempt to adopt the knowledge distillation technique on KGE methods to achieve high performance with low spatial complexity. MulDE [24] is the first work introducing the KD technique on knowledge graph embedding and achieving better performance. Then, DualDE [25] considers the teacher and student model learning together for dynamically transferring more proper knowledge. IterDE [64] proposes a soft-label weighting dynamic adjustment mechanism that can balance the inconsistency of optimization direction between hard and soft label loss by gradually increasing the weighting of soft label loss. Zhu et al. [65] employ knowledge distillation for feature distribution and reception in federated learning. Liu et al. [66] employed a distillation framework for incremental knowledge graph representation, addressing the heterogeneity between new and old triple data while preserving the existing node embedding structure avoiding the catastrophic forgetting issue.

Existing works overlook the unavailability of teacher models in resource-constrained scenarios and the fusion of model views. This work addresses model view fusion through knowledge distillation to enhance generalization and performance of KGE model. We then explore and develop a novel distillation framework for heterogeneous model views fusion of KGE model, thus improving the performance of the lightweight student KGE model without high-performance teacher KGE model in resource-restriction scenarios.

## III. Preliminaries

### A. Knowledge Graph Definition

The knowledge graph (KG) can be defined as $\mathcal{G} \subseteq \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{E}$ and $\mathcal{R}$ represent the sets of entities and relations, respectively. The set of facts (triplets) is $\mathcal{T} \subseteq \{\mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$. Each triplet $(h, r, t) \in \mathcal{T}$, where $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$, denotes a relational mapping from entity $h$ to entity $t$ with relation $r$.

Due to the typical incompleteness of knowledge graphs, knowledge graph embedding methods, that maps entities into continuous low-dimension numerical space and models relation as some transformation, are employed to address this issue via link prediction. Specifically, the task of link prediction is framed as a ranking problem based on scores generated by the scoring function $\Phi : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \to \mathbb{R}$, which takes as input the embeddings of entities $h, t$ and relation $r$. For a given query $(h, r, \cdot)$ and tail entity $\varsigma$ is ground truth, scores are calculated by substituting the tail entity with all other entities from $\mathcal{E}$. These scores are then ranked in descending order, and the position of the tail entity $\varsigma$ is used for model evaluation.

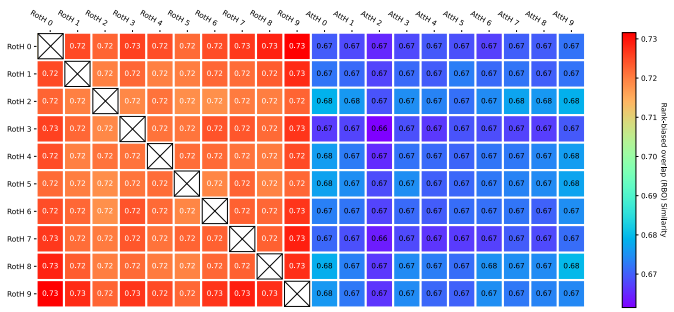


Fig. 3. The similarities of entity ranks produced by AttH and RotH.

### B. Optimization of Knowledge Graph Embedding

Optimizing knowledge graph embeddings is akin to training via contrastive learning. For a given triplet $(h, r, t) \in \mathcal{T}$ (positive sample), we employ a random negative sampling strategy to create a negative sample set $\mathcal{T}' = N((h, r, t); n) = \{(h, r, e_i) \mid e_i \in \mathcal{E}, e_i \neq t, i = 1, \ldots, n\}$, where $N$ is the negative sampling function and $n$ is the number of negative samples generated by replacing the tail entity with other entities.

Using the KGE model, we generate scores for positive and negative triplets: $s = \Phi(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}), (h, r, t) \in \mathcal{T}$ and $s' = \Phi(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}')$, where $(h, r, t') \in \mathcal{T}'$. We then apply a loss function (e.g., BCE loss, Margin loss) to maximize the scores of positive triplets and minimize the scores of negative triplets as follows

$$\min_{\boldsymbol{e} \in \mathcal{E}, \boldsymbol{r} \in \mathcal{R}} -f(s) + \frac{1}{|\mathcal{T}'|} \sum_{(h,r,t') \sim \mathcal{T}'} f(s'), \tag{1}$$

### C. Quantification of Model View Discrepancy

To enhance the model's generalization ability, we propose fusing different model views into a single model. Figure 2 illustrates the difference in entity embedding distributions learned by RotH and AttH, attributed to their distinct model views. To quantify this difference, we introduce an entity ranking method. If the ranking lists of AttH and RotH for the same query triplet are similar, their entity embedding distributions are considered similar, and vice versa.

Specifically, for a given query triplet $(h, r, \varsigma)$, a KGE model is utilized to generated the predicted head entity $\boldsymbol{h}'$, i.e. $\boldsymbol{h}'_{\text{AttH}} = \text{AttH}(\boldsymbol{h}, \boldsymbol{r})$, $\boldsymbol{h}'_{\text{RotH}} = \text{RotH}(\boldsymbol{h}, \boldsymbol{r})$. Then, the scores between the predicted head entity $\boldsymbol{h}'$ and other entities are calculated and these entities are ranked accordingly. We have

$\mathrm{Rank}_{\mathrm{AttH}}(h,r,\cdot) = [\boldsymbol{e}_{\mathrm{a}}, \boldsymbol{e}_{\mathrm{b}}, \boldsymbol{e}_{\mathrm{c}}, \cdots]$, $\mathrm{Rank}_{\mathrm{RotH}}(h,r,\cdot) = [\boldsymbol{e}_{\mathrm{c}}, \boldsymbol{e}_{\mathrm{b}}, \boldsymbol{e}_{\mathrm{f}}, \cdots]$. Finally, a ranking similarity evaluation protocol, named the Rank Biased Overlap (RBO) [67], is used to calculate the similarity between the ranking lists generated by these two KGE models for the same query, i.e. $\mathrm{RBO}\left(\mathrm{Rank}_{\mathrm{AttH}}(h,r,\cdot), \mathrm{Rank}_{\mathrm{RotH}}(h,r,\cdot)\right)$. In addition, to avoid the interference caused by random noise, we train ten RotH and AttH models with distinct random seeds, respectively, and evaluated them on the test set of the FB15K-237. The similarities are shown in Figure 3. Note that the similarity between any two RotH models is higher than the similarity between RotH and AttH. This disparity in ranking similarity indicates the substantial influence of relational mapping methods on learned entity embedding distributions.

### D. Metric Function on Poincaré Space

The hyperbolic space is a Riemannian manifold with constant negative curvature $c < 0$, and Poincaré space $\mathbb{H}$ is one of its five isometric models [39], [68], [69]. And the Poincaré model states that any point in the $d$-dimensional hyperbolic space can be mapped into the $(d-1)$-dimensional spherical space through stereographic projection, so the Poincare space is a bounded spherical space. The metric function mainly measures the distance of any two points in space. In Euclidean space/flatten space $\mathbb{E}$, the metric function can be defined as

$$f_{\mathbb{E}}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^{\top}(\boldsymbol{x}-\boldsymbol{y})}, \tag{2}$$

where $\boldsymbol{x}, \boldsymbol{y}$ are any points in Euclidean space. And for given any two point $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{H}_c$ in Poincaré space with negative curvature $c$, the geodesic distance between them can be defined as

$$f_{\mathbb{H}}(\boldsymbol{x}, \boldsymbol{y}; c) = \frac{2}{\sqrt{|c|}} \tanh^{-1}\left(\sqrt{|c|}\, \| -\boldsymbol{x} \oplus_c \boldsymbol{y} \|_2\right), \tag{3}$$

where the $\oplus_c$ is Möbius addition.

### E. Knowledge Distillation

**Teacher-Student Distillation [70].** The teacher-student architecture is a typical framework in knowledge distillation, used for transferring effective feature information from the teacher model to the student model for model compression and knowledge transfer. Hinton et al. [22] and Ba and Caruana [71] propose to shift the knowledge from teacher network to student network by learning the class distribution via softened, which is termed as response-based distillation. A common way for distillation is optimizing the distributions discrepancy between student and teacher using Kullback-Leibler divergence.

Apart from distilling knowledge from the softened labels, Romero et al. [72] initially introduce hint learning. A hint is defined as the outputs of a teacher's hidden layer, which helps guide the student's learning process. The goal of student learning is to learn a feature representation that is the optimal prediction of the teacher's intermediate representations, which is termed as feature-based distillation.

**Mutual Distillation.** The mutual distillation [73] process begins with a pool of untrained students learning the task together. Each student is trained with two losses: a conventional supervised learning loss and a mimicry loss aligning each student's class posterior with the class probabilities of others. This peer-teaching approach results in each student learning significantly better than in a conventional supervised learning scenario.

## IV. Method

This section introduces **D**ual-view **Mut**ual **D**istillation framework for Knowledge Graph **E**mbedding (DMutDE), a novel knowledge distillation framework designed to fuse two distinct KGE model views into a single model, enhancing its generalization and performance. The overview architecture of the DMutDE is shown in Figure 4. Notably, it eliminates the need of large and high-performance teacher models. Specifically, our framework includes two modules: (1) A novel soft-label fusion (SLF) module to filter interference noise at initial training stage. (2) An entity embedding distillation (EED) module using an encoder to project entity embeddings into learning space, thereby enabling to learn their distributional features adaptively.

### A. An Overview of the DMutDE

In traditional knowledge graph distillation frameworks, the student and teacher models are identical, except the teacher has higher spatial complexity and is pretrained. Consequently, the student model focuses only on learning from the teacher. In contrast, our framework employs two distinct peer KGE models, one student model and one auxiliary model: the student model $M_S$ is for enhancing performance and the auxiliary model $M_A$ is for providing a heterogeneous model view. Both models possess identical embedding dimensionality and are randomly initialized before training. During optimization, they need to use hard labels to learn proper entity and relation embeddings while transferring feature representations through distillation losses to each other, thus integrating each other's model views and improving embedding learning mutually.

**Hard Label Loss.** This loss function is used to optimize the original KGE problem, as shown in Eq.1. In this work, we use the Log-Sigmoid function to reformulate the problem as below

$$L_{\mathrm{Hard}} = -\sum_{(h,r,t)\in\mathcal{T}} \left( \log \sigma(s) + \sum_{(h,r,t')\sim\mathcal{T}'} \log \sigma(-s') \right), \tag{4}$$

where the $\sigma$ is the Sigmoid function, and the $s$, and $s'$ are the scores predicted by positive and negative triplets respectively.

### B. Soft Label Fusion (SLF)

The method for modeling relational mappings determines the model view. Discrepancies between two model views are quantified by evaluating their entity ranking patterns by RBO. Therefore, we can distill the student model with the entities ranking predicted by the auxiliary model as soft labels, thus implementing the dual model view fusion. However, significant noise exists in the entities ranking list predicted

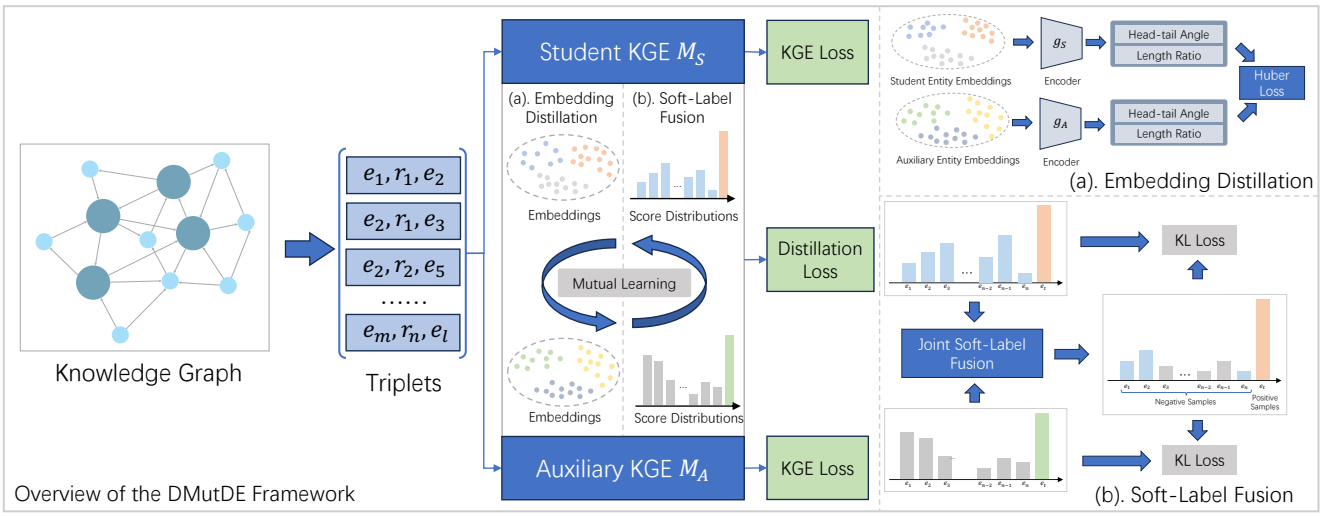


Fig. 4. The left figure is the overview of DMutDE framework. There are two KGE model employed in the framework for distillation with each others. The upper-right figure demonstrates the details of the entity embedding distillation module. And the bottom-right figure illustrates the process of soft-label fusion module.

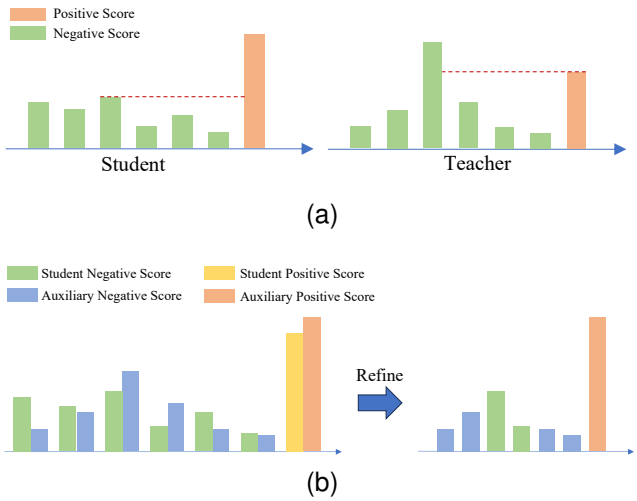


Fig. 5. (a) Illustrating how the noise of the teacher model interferes with the student model. (b) The process of refining the target soft labels by jointly fusing the distributions of student and auxiliary models.

by the auxiliary model in the initial training stage. Directly distilling student model leads to performance degradation. Therefore, we propose a Soft Label Fusion module for refining the soft label.

We denote the score functions of the student model and auxiliary model as $\Phi_S$ and $\Phi_A$ respectively. In the training process, we sample one positive triplet $(h, r, t) \in \mathcal{T}$ from the KG triplets set and generate several associating negative triplets by replacing tail entity with other entities as $(h, r, t') \in \mathcal{T}'$ to form a training subset $T = \{(h, r, t), (h, r, t'_1), \cdots, (h, r, t'_n)\}$. Afterward, by feeding the training subset $T$ into the score functions $\Phi_S$ and $\Phi_A$, we have two score distributions $\mathcal{P}_S = \Phi_S(T)$ and $\mathcal{P}_A = \Phi_A(T)$.

The distillation process for the student model aims to minimize the distributional discrepancy between the student and auxiliary models, typically measured using the Kullback-Leibler (KL) divergence $\mathrm{KL}(\mathcal{P}_S, \mathcal{P}_A)$. However, the presence of noise in the auxiliary model can result in negative scores surpassing positive scores in soft labels, potentially distorting the student model's embeddings (as depicted in Figure 5a). To address this, we first generate new soft labels by jointly fusing the two distributions to filter out these noises, thereby mitigating the impact of inference noise on the student model. Specifically, as the goal of the KGE methods optimization is to maximize the positive scores and minimize the negative scores, we firstly normalize and compare the score distributions predicted by the student model and auxiliary model. And then, we select the smaller negative score and the larger positive score as the final soft label, illustrated in Figure 5b, and mathematically expressed as follows:

$$f_{\text{fusion}}(\tilde{s}_i^S, \tilde{s}_i^A) = \begin{cases} \max\{\tilde{s}_i^S, \tilde{s}_i^A\}, & \tilde{s}_i^S, \tilde{s}_i^A \text{ are positive scores} \\ \min\{\tilde{s}_i^S, \tilde{s}_i^A\}, & \tilde{s}_i^S, \tilde{s}_i^A \text{ are negative scores} \end{cases}, \tag{5}$$

where the $\tilde{s}_i^S = f_{\text{norm}}(\mathcal{P}_S)_i$, $\tilde{s}_i^A = f_{\text{norm}}(\mathcal{P}_A)_i$ are the $i^{\text{th}}$ scores in the normalized score distributions predicted by student model and auxiliary model for one given training subset $T$, $f_{\text{norm}}(\cdot)$ is normalization function. The refined soft-label is denoted as $\tilde{\mathcal{P}} = f_{\text{fusion}}(\tilde{\mathcal{P}}_S, \tilde{\mathcal{P}}_A)$, where $\tilde{\mathcal{P}}_S = f_{\text{norm}}(\mathcal{P}_S)$, $\tilde{\mathcal{P}}_A = f_{\text{norm}}(\mathcal{P}_A)$. Therefore, the distillation loss for student model is $L_{\text{Soft}}^{\text{coarse}} = \mathrm{KL}(\mathcal{P}_S, \tilde{\mathcal{P}})$.

As the score distributions $\mathcal{P}_S$ and $\mathcal{P}_A$ contain both positive and negative scores, the distillation loss $L_{\text{Soft}}^{\text{coarse}}$ performs coarse-grained distillation, transferring knowledge of both positive and negative scores. During optimization, negative sample scores become suppressed to a narrow range after normalization due to the much higher positive sample scores, reducing the effectiveness of transferring knowledge in the distillation process. To address this issue, we remove the positive scores from these distributions, resulting in $\mathcal{P}_S^-$ and $\mathcal{P}_A^-$ for the student model and auxiliary model. We then apply

a re-weighting mechanism to rescale the negative scores as below

$$\mathcal{P}_i^{-\prime} = f_\alpha(\mathcal{P}_i^-) = \frac{\exp(\alpha \cdot \mathcal{P}_i^-)}{\sum_j \exp(\alpha \cdot \mathcal{P}_j^-))} \cdot \mathcal{P}_i^-, i \in \{S, A\} \quad (6)$$

where the $\alpha$ is a learnable smoothing factor. Thus, the re-weighted negative distributions of the student and auxiliary models are $\mathcal{P}_S^{-\prime} = f_\alpha(\mathcal{P}_S^-)$ and $\mathcal{P}_A^{-\prime} = f_\alpha(\mathcal{P}_A^-)$, respectively. The fine-grained distillation loss is then defined as $L_{\text{Soft}}^{\text{fine}} = \text{KL}(f_{\text{norm}}(\mathcal{P}_S^{-\prime}), f_{\text{norm}}(\mathcal{P}_A^{-\prime}))$, focusing solely on distilling feature representations from the negative score distributions. Therefore, the soft-label distillation loss is defined as

$$L_{\text{Sotf}} = (1 - \omega_1) \cdot L_{\text{Soft}}^{\text{coarse}} + \omega_1 \cdot L_{\text{Soft}}^{\text{fine}}, \quad (7)$$

where the $\omega_1$ is hyperparameter.

### C. Entity Embeddings Distillation

The aforementioned knowledge distillation is response-based. To better integrate the model views of the two peer KGE models, we employ a feature-based distillation module to transfer knowledge through entity embeddings. In the previous works [25], the student model directly learns the length ratio and the angle between the head and tail entities. However, due to the heterogeneity of the two peer KGE models in our framework, conventional feature distillation degrades the student's learning. In this work, we use a two-layer neural network $g : \mathbb{R}^d \to \mathbb{R}^{d'}$ as an encoder to project the entity embeddings into a common $d'$-dimensional learning space, and the encoder $g$ is defined as

$$g(\boldsymbol{e}) = \boldsymbol{W}_2 \cdot \sigma(\boldsymbol{W}_1 \cdot \boldsymbol{e} + \boldsymbol{b}_1) + \boldsymbol{b}_2, \quad (8)$$

where the $\sigma$ is activate function, $\boldsymbol{e}$ is entity embedding. Then, we utilize the encoder $g_S$ and $g_A$ for encoding the learned entity embeddings of student and auxiliary models as below.

$$\boldsymbol{e}'_S = g_S(\boldsymbol{e}_S), \ \boldsymbol{e}'_A = g_A(\boldsymbol{e}_A), \quad (9)$$

where the $\boldsymbol{e}_S$ and $\boldsymbol{e}_A$ are the learned entity embeddings of student model and auxiliary model. After projecting the entity embeddings into the $d'$-dimensional learning space, for reflecting the distribution structures of entity embeddings, the length ratio and angle are used. For the encoded head $\boldsymbol{h}'$ and tail entity embeddings $\boldsymbol{t}'$, the length ratio $\varphi_{\text{LR}}$ and angle $\varphi_{\text{angle}}$ are defined as below

$$\varphi_{\text{LR}}(\boldsymbol{h}', \boldsymbol{t}') = \frac{\|\boldsymbol{h}'\|_2}{\|\boldsymbol{t}'\|_2}, \varphi_{\text{angle}}(\boldsymbol{h}', \boldsymbol{t}') = \left\langle \frac{\boldsymbol{h}'}{\|\boldsymbol{h}'\|_2}, \frac{\boldsymbol{t}'}{\|\boldsymbol{t}'\|_2} \right\rangle, \quad (10)$$

Then, for a given triplet $(h, r, t)$, the structure distillation loss $L_{\text{Struct}}$ for entity embeddings is defined as

$$\begin{aligned} L_{\text{Struct}} = &(1 - \omega_2) \cdot l_\delta(\varphi_{\text{LR}}(\boldsymbol{h}'_S, \boldsymbol{t}'_S), \varphi_{\text{LR}}(\boldsymbol{h}'_A, \boldsymbol{t}'_A)) \\ &+ \omega_2 \cdot l_\delta(\varphi_{\text{angle}}(\boldsymbol{h}'_S, \boldsymbol{t}'_S), \varphi_{\text{angle}}(\boldsymbol{h}'_A, \boldsymbol{t}'_A)), \end{aligned} \quad (11)$$

where the $\omega_2$ is a hyperparameter and metric function $l_\delta$ is Huber Loss [74] with $\delta = 1$ defined as

$$l_{\delta=1}(a, b) = \begin{cases} \frac{1}{2}(a-b)^2 & |a-b| \le 1 \\ |a-b| - \frac{1}{2}, & |a-b| > 1 \end{cases}. \quad (12)$$

**Algorithm 1** The Optimization Process of DMutDE

**Input:** KGs $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$;
Student model $M_S$; Auxiliary model $M_A$;
Negative sampling strategy $N(\cdot; n)$ with negative sample size $n$;
Tunable parameters $\mu_1, \mu_2, \eta$;
Epoch number $L$; Distillation epoch number $\theta$.
**Output:** Embeddings of entities and relations in student and auxiliary models.

Randomly initialize the parameters of the student model, auxiliary model, and two embedding encoders $g_S, g_A$.
**for** $i = 1, 2, \cdots, L$ **do**
  **while** sample one triplet $(h, r, t) \sim \mathcal{T}$ **do**
    Construct negative triplets as $\mathcal{T}' = N((h, r, t); n)$.
    *// Student Training Phase ...*
    $L^S = L_{\text{Hard}}^S$.
    **if** $i > \theta$ **then**
      $L^S = L^S + \mu_1 \cdot L_{\text{Soft}}^S + \mu_2 \cdot L_{\text{Struct}}^S$.
    **end if**
    $M_S \leftarrow M_S - \eta \cdot \nabla_{M_S} L^S$.
    *// Auxiliary Training Phase ...*
    $L^A = L_{\text{Hard}}^A$.
    **if** $i > \theta$ **then**
      $L^A = L^A + \mu_1 \cdot L_{\text{Soft}}^A + \mu_2 \cdot L_{\text{Struct}}^A$.
    **end if**
    $M_A \leftarrow M_A - \eta \cdot \nabla_{M_A} L^A$.
  **end while**
**end for**
**return** $M_S, M_A$

Finally, in the DMutDE framework, the student model and auxiliary model are distilled alternatively, and the optimization loss for each part is defined below

$$L = L_{\text{Hard}} + \mu_1 \cdot L_{\text{Soft}} + \mu_2 \cdot L_{\text{Struct}}, \quad (13)$$

where the $\mu_1, \mu_2$ are tunable parameters. For improved distillation, we first independently train the two models using hard label loss to obtain proper entity and relation embeddings, then starting mutual distillation after the $\theta^{\text{th}}$ epoch. The complete optimization process is detailed in Algorithm 1.

## V. Theoretical Analysis

In this section, we present a theoretical analysis of DMutDE, including complexity analysis, and effectiveness analysis. Table II compares the time and spatial complexities of our framework and other baselines.

### A. Complexity Analysis

Prior to the complexity analysis, let us denote the following: $n$ is the number of negative samples per triplet, $n_e$ and $n_r$ are the number of entities and relations, $d$ is the embedding dimensionality of KGE model, $d_s$ and $d_t$ are the dimensionalities of the student and teacher KGE models, respectively, with $d_t > d_s$, $|T|$ is the number of teacher models, and $k$ is the number of top-$k$ candidates. The time complexity of our framework primarily depends on the negative sample size

TABLE II
COMPLEXITY OF SOME KGE MODELS AND KD FRAMEWORKS

| | Time $\mathcal{O}_{\text{time}}$ | Spatial $\mathcal{O}_{\text{spatial}}$ |
|---|---|---|
| Knowledge Graph Embedding Models | | |
| TransE | $\mathcal{O}(d)$ | $\mathcal{O}(n_e d + n_r d)$ |
| RotatE | $\mathcal{O}(d)$ | $\mathcal{O}(n_e d + n_r d)$ |
| AttH | $\mathcal{O}(d)$ | $\mathcal{O}(n_e d + n_r d + n_r)$ |
| RotH | $\mathcal{O}(d)$ | $\mathcal{O}(n_e d + n_r d + n_r)$ |
| RefH | $\mathcal{O}(d)$ | $\mathcal{O}(n_e d + n_r d + n_r)$ |
| Knowledge Distillation Framework for KGE Models | | |
| BKD | $\mathcal{O}(nd_s + nd_t)$ | $\mathcal{O}((n_e + n_r)(d_s + d_t))$ |
| RKD | $\mathcal{O}(nd_s + nd_t)$ | $\mathcal{O}((n_e + n_r)(d_s + d_t))$ |
| MulDE | $\mathcal{O}(n_e d_s + \lvert T\rvert k d_t)$ | $\mathcal{O}((n_e + n_r)(d_s + \lvert T\rvert d_t) + n_r \lvert T\rvert)$ |
| DualDE | $\mathcal{O}(nd_t + nd_s)$ | $\mathcal{O}((n_e + n_r)(d_s + d_t))$ |
| IterDE | $\mathcal{O}(nd_t + nd_s)$ | $\mathcal{O}((n_e + n_r)(d_s + d_t))$ |
| DMutDE | $\mathcal{O}(nd_s d')$ | $\mathcal{O}((n_e + n_r)d_s + d_s d')$ |

and learning space. We assume the time complexity of the student and auxiliary KGE models both are $\mathcal{O}(d_s)$. The Joint Soft Label Fusion module has a time complexity of $\mathcal{O}(nd_s)$ as it considers both positive and negative triplets. In the Entity Embedding Distillation module, the time complexity of encoder $g$ is $\mathcal{O}(d_s d')$ due to matrix-vector multiplications, which are significantly accelerated by CUDA. The time complexity of $L_{\text{Struct}}$ is $\mathcal{O}(d')$. Hence, the overall time complexity of DMutDE is $\mathcal{O}(nd_s d')$.

The time complexity of our framework is primarily determined by the two lightweight KGE models, $\mathcal{O}(n_e d_s + n_r d_s)$, and the encoder $g$, $\mathcal{O}(d_s d')$. Consequently, the overall spatial complexity of DMutDE is $\mathcal{O}((n_e + n_r)d_s + d_s d')$. Unlike other baselines, our framework excludes the teacher model with $\mathcal{O}((n_e+n_r)d_t)$ complexity, significantly reducing storage requirements and making it more suitable for resource-limited scenarios.

### B. Effectiveness Analysis

This section demonstrates the effectiveness of DMutDE and explains, from an information theory perspective, how the DMutDE framework enhances the performance of the student KGE model. Given two distinct KGE methods $M_1$ and $M_2$ applied to the same KG dataset, we assume the entity embeddings they learn are random variables sampled from distinct probability distributions: $z_1 \sim P_{M_1}(z)$ and $z_2 \sim P_{M_2}(z)$ and the mutual information between these distributions is $I(z_1; z_2) > 0$.

**Lemma 1.** *Minimizing the loss function shown in Eq. 13 is equivalent to maximizing the mutual information $I(z_1; z_2)$.*

*Proof.* The $I(z_1; z_2)$ can be written as $I(z_1; z_2) = H(z_1) - H(z_1|z_2)$, and the $-H(z_1|z_2)$ can be decomposed as below

$$\begin{aligned} -H(z_1|z_2) &= \mathbb{E}_{z_2,z_1}[\log p(z_1|z_2)] \\ &= \mathbb{E}_{z_2}[D_{KL}(p(z_1|z_2)||q(z_1|z_2))] \\ &\quad + \mathbb{E}_{z_2,z_1}[\log q(z_1|z_2)] \\ &\geq \mathbb{E}_{z_2,z_1}[\log q(z_1|z_2)] \end{aligned} \tag{14}$$

Thus, we have $I(z_1; z_2) = H(z_1) - H(z_1|z_2) \geq H(z_1) + \mathbb{E}_{z_2,z_1}[\log q(z_1|z_2)]$. And the term $H(z_1)+\mathbb{E}_{z_2,z_1}[\log q(z_1|z_2)]$ is the lower bound of $I(z_1; z_2)$. Here, we assume the $q(z)$ follows normal distribution, thus we have

$$\begin{aligned} \mathbb{E}_{z_2,z_1}[\log q(z_1|z_2)] &= \mathbb{E}_{z_2,z_1}\left[\log \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{(z_1 - \mu(z_2))^2}{2\sigma^2}\right] \\ &= \mathbb{E}_{z_2,z_1}\left[-\log \sigma - \frac{(z_1 - \mu(z_2))^2}{2\sigma^2}\right] \\ &= -\mathbb{E}_{z_2,z_1}\left[\frac{(z_1 - \mu(z_2))^2}{2\sigma^2}\right] + \text{constant} \end{aligned} \tag{15}$$

Thus, we can maximize the lower bound $-\mathbb{E}_{z_2,z_1}\left[(z_1 - \mu(z_2))^2/(2\sigma^2)\right] \leq 0$ to maximize the mutual information $I(z_1; z_2)$ between KGE model $M_1, M_2$. We further generalize this term as $-\mathbb{E}_{z_2,z_1}\left[f_{\text{dist}}(z_1, \mu(z_2))\right]$, where the $\sigma$ can be omitted as a constant. Finally, we reformulate the lower bound of $I(z_1; z_2)$ as below

$$I(z_1; z_2) \geq -\mathbb{E}_{z_2,z_1}\left[f_{\text{dist}}(z_1, \mu(z_2))\right] + \text{constant} \tag{16}$$

We assume the term $\mu$ is a learnable encoder and $f_{\text{dist}}$ is $l_{\delta=1}$, which is same as the entity embedding distillation.

On the other hand, Ahn et al. [63] proof this low bound is still hold for the activations of the intermediate layers, i.e. $\boldsymbol{t}^{(k)} = \mathcal{T}^{(k)}(\boldsymbol{x})$. Here, the score distributions $\mathcal{P}_{M_1} = \Phi_{M1}(T), \mathcal{P}_{M_2} = \Phi_{M_2}(T)$ can be treated as the output of the neural network $\Phi$. Thus, when the $\mu$ is identical mapping and $f_{\text{dist}}$ is KL divergence $D_{\text{KL}}$, then minimizing the loss in the joint soft label fusion module shown in Eq. 7 is equivalent to maximizing $I(z_1; z_2)$. □

**Theorem 1.** *Assume the ideal entity embedding $\tilde{z}$, which can fully portray the structure features of KG data, is drawed from one probability distribution $\tilde{z} \sim p(\tilde{z})$. The optimization of DMutDE mainly maximizes the mutual information $I(z_1; \tilde{z})$ and $I(z_2; \tilde{z})$.*

*Proof.* We denote the entity embeddings learned by two distinct KGE methods $M_1, M_2$ as $z_1 \sim p_{M_1}(z_1), z_2 \sim p_{M_2}(z_2)$, and assume an ideal entity embeddings $\tilde{z} \sim p(\tilde{z})$ that can fully portray the structure features of KG data exist. In addition, some structures are relatively easy to be represented, thus we have $I(z_1; \tilde{z}), I(z_2; \tilde{z}), I(z_1; z_2; \tilde{z}) > 0$. Then, the mutual information $I(z_1 z_2; \tilde{z})$ between joint $z_1, z_2$ and $\tilde{z}$ indicates the total volume of the effective structure features that the two KGE models learned. By the chain rule of mutual information, we have

$$\begin{aligned} I(z_1 z_2; \tilde{z}) &= I(z_1 l\tilde{z}) + I(z_2; \tilde{z}|z_1) \\ &= I(z_1; \tilde{z}) - I(z_1; z_2; \tilde{z}) + I(z_2; \tilde{z}) \\ &= I(z_1; \tilde{z}) + I(z_2; \tilde{z}) - I(z_1; z_2; \tilde{z}) \\ &= I(z_1; \tilde{z}) + I(z_2; \tilde{z}) - [I(z_1; z_2) - I(z_1; z_2|\tilde{z})] \\ &= I(z_1; \tilde{z}) + I(z_2; \tilde{z}) - I(z_1; z_2) + I(z_1; z_2|\tilde{z}) \end{aligned} \tag{17}$$

where, $I(z_1; \tilde{z})$ and $I(z_2; \tilde{z})$ are the effective features leaned by KGE model $M_1$ and $M_1$ respectively. $I(z_1; z_2)$ is the common effective features learned these two KGE model and $I(z_1; z_2|\tilde{z})$ is the common ineffective features that these KGE models learned.

In the process of optimizing these two KGE models, since no other distinct model view is introduced, the mutual information $I(z_1 z_2; \tilde{z})$ between them and the ideal model remains unchanged regardless of whether DMutDE is used to distill them. After mutual distillation using DMutDE, according to Lemma 1, $-I(z_1; z_2)$ is decreased and other terms have to be increased. On the other hand, the term $I(z_1; z_2|\tilde{z})$ contains noise/superfluous embedding information, and these noise generally interfere normal embeddings and produce abnormal triplets scores, e.g. positive triplets scores are less than negative triplets scores. The filter function $f_{\text{fusion}}$ as shown in Eq. 5 in joint soft label fusion module becomes a gate mechanism to control the exchanges of the effective information between the two KGE models, and greatly prevents the transmission of harmful information. Finally, with the mutual distillation processing, $I(z_1; \tilde{z}), I(z_2; \tilde{z})$ are increasing considerably. □

The Theorem 1 shows that as the optimization of DMutDE proceeds, the entity embedding distillation module maximizes the mutual information $I(z_1; z_2)$ between the student model and the auxiliary model, while $f_{\text{fused}}$ as a gate mechanism plays a vital role in preventing the propagation of harmful noise and the amplification of noise/invalid information $I(z_1; z_2|\tilde{z})$. According to Equation 17, these two modules facilitate the effective information $I(z_1; \tilde{z})$ of the student model and the effective information $I(z_2; \tilde{z})$ of the auxiliary model continue to increase.

## VI. Experiments

In this section, we evaluate the DMutDE framework's performance on link prediction tasks using two standard public benchmarks, comparing it against several state-of-the-art methods. Comprehensive experiments and analyses are conducted to address three key research questions:

- ***RQ1:*** *Can DMutDE distill a superior student KGE model without large teacher models?*
- ***RQ2:*** *Can any two KGE methods with heterogeneous model view be effectively fused?*
- ***RQ3:*** *What is the contribution of each module?*

We summarize the general experimental settings before presenting the results and model analysis.

### A. Experimental Setup

*1) Benchmark Descriptions:* We used two standard knowledge graphs (KGs) for the knowledge graph completion task: WN18RR [75] and FB15K-237 [49], derived from the real-world knowledge bases WordNet [76] and Freebase [77], respectively. The statistics for these benchmarks are shown in Table III. Gu et al. [18], [44] use the symbol $\xi_G$ to represent the average curvature of the KGs, indicating their topological structures: $\xi_G < 0$ signifies a prevalence of hierarchical (tree-like) structures, while $\xi_G > 0$ indicates numerous ring structures.

- The WN18RR benchmark, a subset of WN18, comprises 11 relations and 40,943 entities, representing word senses and lexical relationships. The relations exhibit logical patterns such as asymmetry, symmetry, composition, and inversion. As shown in Table III, the $\xi_G$ of WN18RR indicates the presence of numerous simple hierarchical subgraphs, e.g., `(car, hypernym_of, sedan)`.
- FB15K-237, a subset of FB15K excluding reversible relations, contains 14,541 entities and 237 different relations, encompassing locations, movies, people, with relations like `directed_by` and `located_at`. Its $\xi_G$ is relatively larger than that of WN18RR, indicating more complex topological connections in FB15K-237, with various subgraph structures (e.g., tree-like, ring-like, and chain-like) nested within this KG.

*2) Evaluation Protocols:* To evaluate the performance of our framework and baselines, we use Mean Reciprocal Rank (MRR) and Hit at Top-K (Hit@K) metrics for evaluating the knowledge graph completion task. For a given test triplet $\{h, r, t\}$, where $t$ is to be predicted, MRR (↑)[1] measures the mean of inverse ranks assigned to correct entities, and Hit@K (↑) ($K = 1, 3, 10$) measures the proportion of correct triples among the top K predictions. During evaluation, true triplets in the training set are filtered out after ranking all entities for the test triplets. The experiments focus on the performance of the student model within the DMutDE framework.

*3) Baseline Descriptions:* To evaluate the effectiveness of our framework, several state-of-the-art knowledge distillation frameworks for knowledge graph embeddings are used in the experiments as baselines, and these are categorized into two branches: (1) The conventional general knowledge distillation frameworks in other fields, marked with a star, e.g. natural language processing (NLP), computer vision (CV), multi-modal representation learning. (2) The KGE model-oriented knowledge distillation frameworks are marked with a dot.

* BKD [22] is the most basic and commonly used responding-based KD method. We use BKD to minimize the KL divergence of the triplet score distributions generated by the teacher and student. In the experiments, we define the optimization loss as $L = L_{\text{KGE}} + \omega_{\text{BKD}} \cdot L_{\text{BKD}}$, where the $\omega_{\text{BKD}}$ is tunable.
* RKD [61] is an early feature-based knowledge distillation (KD) method that minimizes the feature distance between the student and teacher models. In our experiments, the final optimization loss is defined as $L = L_{\text{KGE}} + \omega_{\text{BKD}} \cdot L_{\text{BKD}} + \omega_{\text{RKD}} \cdot L_{\text{RKD}}$. We utilize the length ratio and cosine similarity of the head and tail embeddings to capture the structural relationships in entity embeddings.
* TA [62] introduces an assistant teacher to bridge the gap between the teacher and student, thereby accelerating the distillation process. In our experiments, the embedding dimensions are 512 for the teacher model, 32 for the student model, and 256 for the assistant teacher model, which shares the same architecture as the student. The distillation proceeds by first transferring knowledge from the teacher to the assistant teacher, then from the assistant teacher to the student
* VID [63], motivated by the information theory, proposes a new knowledge distillation named Variational Information Distillation framework (VID) maximizing

[1]↑: The metric with higher score, the model with better performance

TABLE III
STATISTIC INFORMATION OF THREE BENCHMARKS, AVERAGE NODE DEGREE PLUS/MINUS STANDARD DEVIATION. $\xi_G$ IS THE CURVATURE OF GRAPH.

| Dataset | #Entities | #Relations | #Training | #Validation | #Test | Avg. #Degree | Heterogeneity | Scale | $\xi_G$ |
|---|---|---|---|---|---|---|---|---|---|
| WN18RR | 40,943 | 11 | 86,835 | 3,034 | 3,134 | 2.2±3.6 | Low | Small | -2.54 |
| FB15K-237 | 14,541 | 237 | 272,115 | 17,535 | 20,466 | 19.7±30 | High | Medium | -0.65 |

TABLE IV
THE TABLE SHOWS THE PERFORMANCES (MEAN±STD.) OF SEVERAL KGE METHODS WITH EMBEDDING DIMENSION 32 ON THE TWO BENCHMARKS. THE $a\,\mathrm{e}\,b = a \times 10^b$, FOR EXAMPLE, $1\,\mathrm{e}\,{-1} = 1 \times 10^{-1}$

| Space | Methods | WN18RR | | | FB15K-237 | | |
|---|---|---|---|---|---|---|---|
| | | MRR | Hit@1 | Hit@10 | MRR | Hit@1 | Hit@10 |
| $\mathbb{E}$ | TransE | 0.258±9.4 e −4 | 0.103±1.2 e −3 | 0.489±1.0 e −3 | 0.305±4.8 e −4 | 0.216±7.8 e −4 | 0.484±7.1 e −4 |
| $\mathbb{E}$ | RotatE | 0.396±4.9 e −3 | 0.377±3.5 e −3 | 0.427±6.9 e −3 | 0.271±1.5 e −3 | 0.187±1.5 e −3 | 0.438±2.3 e −3 |
| $\mathbb{H}$ | AttH | 0.464±3.4 e −3 | 0.421±3.1 e −3 | 0.541±6.9 e −3 | 0.318±1.8 e −4 | 0.229±1.3 e −3 | 0.498±1.0 e −3 |
| $\mathbb{H}$ | RotH | **0.473±6.4 e −4** | **0.431±1.5 e −4** | **0.553±2.3 e −3** | 0.311±9.7 e −4 | 0.222±1.6 e −3 | 0.488±6.7 e −4 |
| $\mathbb{H}$ | RefH | 0.462±1.0 e −3 | 0.427±7.4 e −4 | 0.529±2.8 e −3 | 0.305±1.3 e −3 | 0.219±1.5 e −3 | 0.479±1.6 e −3 |
| $\mathbb{H}$ | LocAttH | 0.448±4.7 e −3 | 0.409±4.1 e −3 | 0.517±7.6 e −3 | **0.320±3.8 e −4** | **0.230±1.2 e −3** | **0.501±9.3 e −4** |
| $\mathbb{H}$ | LocRotH | 0.453±3.2 e −3 | 0.416±4.0 e −3 | 0.523±1.3 e −3 | 0.314±1.1 e −3 | 0.223±1.2 e −3 | 0.495±1.6 e −3 |
| $\mathbb{H}$ | LocRefH | 0.468±1.2 e −3 | 0.429±1.9 e −3 | 0.543±1.8 e −3 | 0.314±5.1 e −4 | 0.226±9.1 e −4 | 0.492±9.4 e −4 |

the mutual information of student and teacher models. In our experiments, we follow the paradigm of VID and introduce a distillation loss for entity embedding distillation as $L = L_{\mathrm{KGE}} + \omega_{\mathrm{VID}} \cdot L_{\mathrm{VID}}$

- MulDE [24] is the first knowledge distillation framework for KGE, which proposes leveraging multiple high-dimension pre-trained KGE models as teachers to transfer soft labels to a student model.
- DualDE [25] considers the dual influence between the teacher and the student in the distillation process, thus designing a soft label evaluation mechanism to distinguish the quality of soft labels and a two-stage learning manner.
- IterDE [64] designs a new soft-label weighting dynamic adjustment mechanism for accelerating the distillation process by iteratively adjusting the weights between hard-labels and soft-labels.

The baselines share a common characteristic: they all require one or more pre-trained large teacher models to provide high-quality feature representations. Additionally, most baselines use the same model type for both student and teacher models, except for MulDE. For fairness, we design two comparison settings: (i) reducing the embedding dimension of the teacher models to match that of the student models, and (ii) maintaining a high embedding dimension (512) for the teacher models while using a lower dimension (32) for the student models.

There are several state-of-the-art KGE models adopted in our experiments as student model and auxiliary model. TransE [14], RotatE [15] embed entities in Euclidean space $\mathbb{E}$. AttH, RotH, and RefH [18] embed entities in Poincaré space $\mathbb{H}$. Inspired by GIE [47], which introduces a geometric interaction module for modeling KG data from local view, we adapt this module for AttH, RotH, and RefH, creating Loc-AttH, Loc-RotH, and Loc-RefH.

*4) Implementation Details:* For DMutDE framework in the experiments, the hyper-parameters are tuned with grid searching. The $\omega_1, \omega_2$ are search within $[0, 1]$, and the $\mu_1, \mu_2$ are tuned within $[0, 2]$. The embedding dimension of the student model and auxiliary model are selected from $\{32, 512\}$. The distillation starting epoch number $\theta$ is selected from $\{0, 20, 40, 60, 80, 100\}$. The optimizer is selected from {SGD, Adagrad, Adam, AdamW} and the learning rate is selected from $\{5 \times 10^{-4}, 5 \times 10^{-3}, 5 \times 10^{-2}, 1 \times 10^{-1}\}$. The KGE training framework we used is provided by Chami et.al. [18] and the PyTorch [78] deep learning library. The platform that implementing DMutDE framework is a server equipped with Intel i9-14900k and NVIDIA RTX 4090 GPU. The code for our framework is in https://github.com/RuizhouLiu/DMutDE/tree/master. The key hyper-parameters of DMutDE are reported in Table X

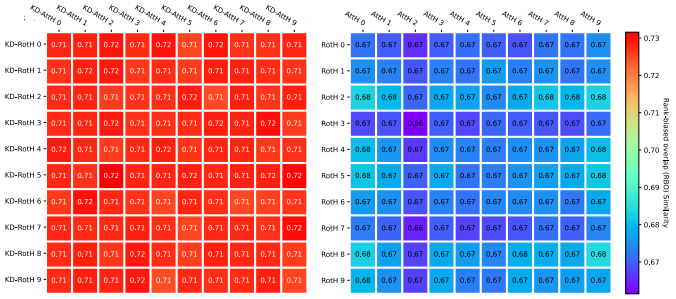

Fig. 6. Visualization of the rank similarity of the Roth and AttH before(right) / after(left) using our framework for distilling feature representations.

## B. Experimental Results

*1) **RQ1:** Can DMutDE distill a superior student KGE model without large teacher models?:* We conduct several comprehensive experiments to evaluate the effectiveness of our framework on two benchmarks with different experimental settings. Firstly, we reproduce these KGE methods of 32 embedding dimensions shown in Table IV with their officially provided code and hyper-parameters [18]. Then, Table V summarizes the performance of the baselines and our framework when the embedding dimensions of both the student model

TABLE V
RESULTS OF THE DMUTDE AND OTHER COMPARED METHODS WITH 32 EMBEDDING DIMENSION FOR STUDENT MODEL. "STU." IS STUDENT MODEL AND "AUX." IS AUXILIARY MODEL. COLUMN-WISE COMPARISON, FIRST BEST IS **BOLD**, SECOND BEST IS UNDERLINED. THE $a\,\mathrm{e}\,b = a \times 10^{b}$, FOR EXAMPLE, $1\,\mathrm{e}\,{-1} = 1 \times 10^{-1}$

| Dataset | KGE Methods | BKD | | | RKD | | | VID | | | DMutDE | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Stu.(Aux.) | MRR | Hit@1 | Hit@10 | MRR | Hit@1 | Hit@10 | MRR | Hit@1 | Hit@10 | MRR | Hit@1 | Hit@10. |
| WN18RR | TransE (TransE) | 0.255±1.4e-3 | 0.104±4.2e-4 | 0.520±2.1e-3 | **0.261±1.2e-3** | 0.115±1.9e-4 | 0.529±1.1e-3 | 0.223±4.1e-4 | 0.055±2.8e-4 | 0.506±6.2e-4 | 0.258±2.8e-4 | 0.107±1.1e-4 | 0.523±4.9e-4 |
| | TransE (RotatE) | 0.256±1.1e-3 | 0.107±2.7e-4 | 0.523±4.1e-3 | 0.258±9.8e-4 | 0.109±3.1e-4 | 0.520±1.9e-3 | 0.241±2.7e-4 | 0.079±8.3e-4 | 0.499±1.4e-3 | **0.263±4.1e-4** | 0.116±2.1e-4 | 0.528±5.7e-4 |
| | RotatE (RotatE) | **0.400±1.3e-3** | 0.370±7.4e-4 | 0.451±3.0e-3 | 0.382±2.1e-3 | 0.350±4.4e-4 | 0.430±2.9e-3 | 0.374±1.5e-3 | 0.349±6.5e-4 | 0.406±2.9e-4 | 0.399±1.0e-3 | 0.373±5.5e-4 | 0.449±3.2e-3 |
| | RotatE (TransE) | 0.399±1.1e-3 | 0.370±4.1e-4 | 0.443±2.7e-3 | 0.387±4.6e-4 | 0.358±2.1e-4 | 0.433±3.0e-3 | 0.368±2.2e-3 | 0.324±4.3e-4 | 0.397±2.1e-3 | **0.402±4.3e-4** | 0.373±9.7e-4 | 0.450±8.2e-3 |
| | AttH (AttH) | 0.455±7.9e-4 | 0.412±5.5e-4 | 0.534±1.4e-3 | 0.451±8.9e-4 | 0.405±6.4e-4 | 0.534±2.1e-3 | 0.286±4.2e-4 | 0.207±2.1e-4 | 0.438±9.7e-4 | **0.463±3.9e-4** | 0.424±1.5e-3 | 0.544±4.1e-3 |
| | AttH (RotH) | 0.460±1.3e-3 | 0.414±6.6e-4 | 0.545±3.5e-3 | 0.456±1.1e-3 | 0.411±5.7e-4 | 0.535±3.9e-3 | 0.275±7.9e-4 | 0.192±4.7e-4 | 0.441±1.2e-3 | **0.475±2.7e-4** | 0.430±7.8e-4 | 0.558±3.8e-3 |
| | AttH (RefH) | 0.460±2.6e-3 | 0.414±9.7e-4 | 0.544±4.8e-3 | 0.449±2.5e-3 | 0.411±1.0e-3 | 0.517±5.1e-3 | 0.259±3.8e-4 | 0.184±4.5e-4 | 0.411±8.4e-4 | **0.477±1.0e-3** | 0.432±1.0e-3 | 0.559±2.6e-3 |
| | RotH (RotH) | 0.471±1.2e-3 | 0.424±8.3e-4 | 0.556±2.4e-3 | 0.465±1.6e-3 | 0.420±5.8e-4 | 0.549±9.4e-4 | 0.286±1.2e-3 | 0.204±3.5e-4 | 0.441±3.2e-3 | **0.474±6.4e-4** | 0.429±2.7e-4 | 0.563±4.9e-3 |
| | RotH (RefH) | 0.474±8.7e-4 | 0.429±5.2e-4 | 0.557±1.8e-3 | 0.453±9.4e-4 | 0.413±7.6e-4 | 0.529±3.7e-3 | 0.267±8.7e-4 | 0.190±4.9e-4 | 0.415±1.6e-3 | **0.480±8.9e-4** | 0.438±5.0e-4 | 0.560±3.3e-3 |
| | RotH(AttH) | 0.469±3.5e-3 | 0.424±1.0e-3 | 0.555±4.1e-3 | 0.462±2.3e-3 | 0.418±9.3e-4 | 0.540±3.4e-3 | 0.293±1.7e-3 | 0.215±5.6e-4 | 0.448±9.8e-4 | **0.478±1.1e-3** | 0.436±3.6e-4 | 0.554±4.0e-3 |
| | RefH (RefH) | 0.448±1.1e-3 | 0.411±9.4e-4 | 0.519±2.8e-3 | 0.446±1.6e-3 | 0.410±5.7e-4 | 0.514±9.4e-4 | 0.184±7.5e-4 | 0.130±4.7e-4 | 0.295±2.4e-3 | **0.450±1.1e-3** | 0.410±7.9e-4 | 0.526±1.6e-3 |
| | RefH (AttH) | 0.450±1.3e-3 | 0.413±4.1e-4 | 0.522±9.3e-4 | 0.452±1.5e-3 | 0.411±4.8e-4 | 0.531±1.7e-3 | 0.183±2.5e-3 | 0.125±8.7e-4 | 0.297±4.5e-3 | **0.454±4.0e-4** | 0.412±1.2e-3 | 0.537±2.2e-3 |
| | RefH (RotH) | 0.452±3.1e-3 | 0.411±2.9e-3 | 0.529±4.7e-3 | 0.451±2.4e-3 | 0.416±1.3e-3 | 0.530±1.8e-3 | 0.168±9.6e-4 | 0.116±1.7e-3 | 0.267±2.0e-3 | **0.454±1.0e-3** | 0.414±2.4e-4 | 0.530±5.7e-3 |
| FB15K-237 | TransE (TransE) | **0.312±7.6e-4** | 0.224±4.1e-4 | 0.487±1.0e-3 | 0.310±1.3e-3 | 0.221±1.5e-3 | 0.488±2.1e-3 | 0.304±1.0e-3 | 0.217±9.3e-4 | 0.479±1.8e-3 | 0.312±9.4e-4 | 0.221±2.8e-4 | 0.488±1.8e-3 |
| | TransE (RotatE) | **0.312±8.4e-4** | 0.223±3.8e-4 | 0.489±9.6e-4 | 0.308±9.8e-4 | 0.220±1.8e-3 | 0.485±1.7e-3 | 0.165±1.2e-3 | 0.106±9.8e-4 | 0.279±1.4e-3 | 0.309±1.6e-3 | 0.223±7.2e-4 | 0.480±2.0e-3 |
| | RotatE (RotatE) | 0.285±1.2e-3 | 0.199±1.5e-3 | 0.458±2.1e-3 | **0.291±6.4e-4** | 0.206±1.0e-3 | 0.459±8.9e-4 | 0.277±1.1e-3 | 0.194±1.5e-3 | 0.443±2.2e-3 | 0.284±6.8e-4 | 0.202±2.3e-4 | 0.448±1.3e-3 |
| | RotatE (TransE) | 0.284±1.3e-3 | 0.197±6.0e-4 | 0.458±2.9e-3 | 0.285±2.1e-3 | 0.200±2.8e-3 | 0.455±4.0e-3 | 0.263±4.7e-4 | 0.187±5.9e-4 | 0.421±6.1e-4 | **0.293±3.4e-3** | 0.205±7.4e-4 | 0.465±5.9e-4 |
| | AttH (AttH) | 0.319±1.4e-3 | 0.229±2.7e-3 | 0.499±1.8e-3 | 0.317±9.1e-4 | 0.226±3.2e-4 | 0.498±5.6e-4 | 0.249±1.4e-3 | 0.167±2.1e-3 | 0.412±2.3e-3 | **0.324±1.8e-4** | 0.233±2.1e-3 | 0.505±1.8e-3 |
| | AttH (RotH) | 0.314±7.7e-4 | 0.227±6.8e-4 | 0.494±1.1e-3 | 0.316±2.1e-3 | 0.226±3.7e-4 | 0.497±1.2e-3 | 0.243±8.9e-4 | 0.163±3.5e-4 | 0.401±8.9e-4 | **0.326±4.0e-4** | 0.234±1.8e-3 | 0.507±2.9e-3 |
| | AttH (RefH) | 0.315±8.6e-4 | 0.226±5.9e-4 | 0.495±1.0e-3 | 0.316±2.3e-3 | 0.226±3.5e-4 | 0.496±1.5e-3 | 0.245±9.2e-4 | 0.166±5.9e-4 | 0.402±7.6e-4 | **0.321±1.3e-3** | 0.232±4.9e-4 | 0.500±7.7e-4 |
| | RotH (RotH) | 0.314±1.4e-3 | 0.225±8.3e-4 | 0.491±5.8e-4 | 0.313±9.5e-4 | 0.225±1.2e-3 | 0.492±2.8e-3 | 0.239±7.5e-4 | 0.158±9.4e-4 | 0.401±1.2e-3 | **0.317±5.2e-4** | 0.225±1.9e-3 | 0.499±8.9e-4 |
| | RotH (RefH) | 0.312±1.0e-3 | 0.223±9.2e-4 | 0.490±8.1e-4 | 0.310±2.0e-3 | 0.223±1.7e-3 | 0.489±1.2e-3 | 0.246±1.1e-3 | 0.163±2.3e-3 | 0.408±1.6e-3 | **0.315±1.3e-4** | 0.224±3.1e-4 | 0.497±3.6e-4 |
| | RotH (AttH) | 0.318±1.2e-3 | 0.228±1.5e-3 | 0.496±2.7e-3 | 0.317±1.8e-3 | 0.227±2.1e-3 | 0.493±2.8e-3 | 0.250±7.7e-4 | 0.172±1.1e-3 | 0.404±1.5e-3 | **0.323±1.0e-3** | 0.232±5.9e-4 | 0.504±2.5e-3 |
| | RefH (RefH) | 0.305±8.9e-4 | 0.218±9.7e-4 | 0.482±1.1e-3 | **0.312±1.1e-3** | 0.225±9.5e-4 | 0.487±1.8e-3 | 0.221±1.1e-3 | 0.149±8.2e-4 | 0.362±2.1e-3 | 0.308±5.1e-4 | 0.220±1.3e-3 | 0.484±3.8e-3 |
| | RefH (AttH) | 0.313±9.7e-4 | 0.226±1.3e-3 | 0.488±2.8e-3 | 0.306±8.8e-4 | 0.220±1.9e-3 | 0.480±1.5e-3 | 0.223±1.2e-3 | 0.156±1.0e-3 | 0.357±1.7e-3 | **0.316±6.2e-4** | 0.227±1.2e-4 | 0.494±4.8e-3 |
| | RefH (RotH) | 0.306±1.8e-3 | 0.219±9.9e-4 | 0.481±1.2e-3 | 0.303±3.1e-3 | 0.215±2.2e-3 | 0.479±2.4e-3 | 0.224±1.7e-3 | 0.156±2.2e-3 | 0.360±3.7e-3 | **0.313±1.5e-4** | 0.225±1.4e-4 | 0.491±5.6e-4 |

and auxiliary model are 32. The following observations are obtained.

1. DMutDE framework can considerably promote the performance of KGE models when the embedding dimensions of the two peer models are the same. Since the embedding dimension of the student model and auxiliary model are identical (both are 32), their capabilities for capturing topological features are approximately the same. Thus, compared with the KGE methods without knowledge distillation shown in Table IV, the performance gains brought by the BKD and RKD are relatively ignored, and some results are even degraded. The performance of DMutDE proves that it can successfully fuse the different views of another KGE model and obtain significant gains. In the FB15K-237, comparing with the results of the KGE methods without knowledge distillation in Table IV, DMutDE improves the MRR of AttH, RotH and RefH from 0.318 to 0.324 (1.8%), from 0.311 to 0.324 (4.2%), from 0.304 to 0.316 (4.0%) respectively. In summary, the average gain of DMutDE is 3.33%, 3.60% and 2.77% on MRR, Hit@1 and Hit@10. While the average gains of other baselines BKD, RKD and VID, on MRR, are 0.43%, 0.37% and -0.76%. VID degrades the student model's performance because it cannot distinguish beneficial heterogeneity information in the KGE model from harmful heterogeneity information that interferes with the distillation process. The performance gains of our framework are considerably more highlighted than other baselines. Furthermore, it is worth noting that although M-DualDE-RotH [25] distilled by multiple 512-dimensional teacher models, achieves 0.328 on MRR, which only 4‰ higher than that of RotH distilled by DMutDE, it must affords more costs on training multiple teacher models.

In WN18RR, compared with the KGE methods without knowledge distillations, the average improvement of DMutDE on MRR is 2.1%. Through the distillation of DMutDE, the performance of the AttH gains from 0.460 to 0.477 (3.7%) on MRR, from 0.421 to 0.432 (2.6%) on Hit@1 and from 0.529 to 0.559 (5.7%) on Hit@10. However, since the limitation of embedding dimension, the teacher model cannot bring large gains for the student model, thus the improvements of BKD and RKD are negligible. In contrast, the gains distilled by DMutDE for AttH and RotH are distinct. In addition, the M-DualDE-RotH is only higher 3‰ than Roth distilled by DMutDE in the WN18RR.

2. From the perspective of entities ranking similarity, we can observe that our framework successfully fuses different model views into one KGE model. Here, we compare the ranking similarity of RotH and AttH with/without the DMutDE. As Figure 6 shows, the elements in the left matrix are the rank similarity of the RotH and AttH that are distilled with the DMutDE framework, while the elements in the right matrix are the rank similarity of the vanilla AttH and RotH trained individually. As we can see, through distillation of effective feature representations within our framework, their ranking similarity raised from 0.67 to 0.71 compared with the right matrix, which is closed with rank similarity of any two vanilla RotHs, see the left matrix in Figure 3 .

*2) **RQ2:** Can any two KGE methods with heterogeneous model view be effectively fused?:* To answer the question, we conduct four case studies to explore the limitation of our framework.

TABLE VI
RESULTS OF GLOBAL-VIEW AND LOCAL-VIEW FUSION WITH THE DMUTDE."STU." IS STUDENT MODEL AND "AUX." IS AUXILIARY MODEL.

| Dataset | KGE Methods Stu. (Aux.) | DMutDE | | | |
|---|---|---|---|---|---|
| | | MRR | Hit@1 | Hit@3 | Hit@10 |
| WN18RR | AttH (LocAttH) | 0.465±5.9e-4 | 0.424±7.3e-4 | 0.480±1.2e-3 | 0.545±9.8e-4 |
| | RotH (LocRotH) | **0.479±1.0e-3** | **0.437±1.4e-3** | **0.493±1.6e-3** | **0.558±2.1e-3** |
| | RefH (LocRefH) | 0.453±1.1e-3 | 0.413±2.5e-3 | 0.466±1.9e-3 | 0.527±2.8e-3 |
| | LocAttH (AttH) | 0.470±8.6e-4 | 0.429±6.0e-4 | 0.488±1.0e-3 | 0.544±2.5e-3 |
| | LocRotH (RotH) | 0.474±2.7e-3 | 0.433±1.9e-3 | 0.492±2.5e-3 | 0.547±2.1e-3 |
| | LocRefH (RefH) | 0.467±8.6e-4 | 0.428±9.7e-4 | 0.480±1.0e-3 | 0.542±5.6e-4 |
| FB15K-237 | AttH (LocAttH) | 0.324±8.8e-4 | 0.232±1.2e-3 | 0.357±1.9e-3 | 0.506±1.0e-3 |
| | RotH (LocRotH) | 0.317±6.8e-4 | 0.225±7.1e-4 | 0.351±1.0e-3 | 0.502±1.3e-3 |
| | RefH (LocRefH) | 0.310±2.8e-3 | 0.221±1.4e-3 | 0.338±9.7e-4 | 0.487±3.2e-3 |
| | LocAttH (AttH) | **0.326±5.8e-4** | **0.234±6.2e-4** | **0.359±1.1e-3** | **0.509±9.6e-4** |
| | LocRotH (RotH) | 0.321±9.1e-4 | 0.228±8.7e-4 | 0.356±2.9e-3 | **0.509±1.4e-3** |
| | LocRefH (RefH) | 0.315±1.1e-3 | 0.227±7.2e-4 | 0.343±9.3e-4 | 0.493±1.4e-3 |

**Case Study 1** In the KGE methods, the discrepancy of manners in modeling relational mapping influences the model views the KGE model for modeling KG data, resulting in different rankings of entities predicted by the model. Therefore, this case study mainly discusses whether the DMutDE can fuse different model views to boost the generalization of a single KGE model. Here, we use the AttH, RotH, and RefH as basic KGE methods for distillation since their measure function and embedding space are identical except for relation-specific geometric transformations. The experimental results are shown in Table V. Through observing the experimental results, the DMutDE can distill a better student model with the discrepancy of relation-specific transformations.

**Case Study 2** The conventional Poincaré space-based KGE methods (e.g. AttH, RotH, and RefH) usually embed entities into single hyperbolic space, which results in the space curvature of modeling topological features surrounding entities is consistent. Therefore, these KGE methods are named as global view KGE methods. Inspired by the GIE [47], we can first map entities into multiple hyperbolic spaces with distinct curvatures, and then the geometric interaction module is employed to fuse the learned embeddings from different spaces to adaptively produce entity embeddings. In this way, the KGE can select the topological features adaptively from multiple hyperbolic spaces. Thus, these KGE methods that employing the geometric interaction can model KGs data from local views. We propose a Local-view framework for general Poincaré space-based KGE methods. We embed the AttH, RotH, and RefH into the local-view framework and create LocAttH, LocRotH and LocRefH respectively.

In this case study, we mainly discuss whether DMutDE can distill a better student model by fusing the global-view and local-view KGE methods together. Here, the pairs of (student model and auxiliary model) are (LocKGE, KGE) or (KGE, LocKGE), and the experimental results are shown as Table VI. Comparing with the original KGE model, the average gains of the local-view KGE models are 2.83% on MRR, 2.84% on Hit@1, 2.71% on Hit@3 and 2.44% on Hit@10, while the global-view KGE model average increase 1.01%, 1.03%, 0.61%, 0.93% on MRR, Hit@1, Hit@3 and Hit@10 in the WN18RR benchmark. In FB15K-237, the average increases are 1.26% and 1.94% on the MRR for the local-view model and global-view model. Therefore, the fusing the global and local views with our framework can boost the performance of these KGE models. In addition, the LocKGE models possess more potentials to comprehensively learn the effective feature representations from the global-view.

TABLE VII
RESULTS OF FUSING KGE MODELS WITH HETEROGENEOUS MEASURE SPACES."STU." IS STUDENT MODEL AND "AUX." IS AUXILIARY MODEL. $\mathbb{H}$ ($\mathbb{E}$) MEANS STUDENT MODEL IS IN POINCARÉ SPACE AND AUXILIARY MODEL IS IN EUCLIDEAN SPACE.

| Dataset | KGE Methods Stu. (Aux.) | Space | DMutDE | | | |
|---|---|---|---|---|---|---|
| | | | MRR | Hit@1 | Hit@3 | Hit@10 |
| WN18RR | AttH (TransE) | $\mathbb{H}$ ($\mathbb{E}$) | 0.462±2.4e-3 | 0.419±1.7e-3 | 0.478±2.2e-3 | 0.540±3.1e-3 |
| | AttH (RotatE) | $\mathbb{H}$ ($\mathbb{E}$) | 0.466±1.5e-3 | 0.424±1.1e-3 | 0.483±7.9e-4 | 0.545±1.9e-3 |
| | RotH (TransE) | $\mathbb{H}$ ($\mathbb{E}$) | 0.468±3.2e-3 | 0.425±8.2e-4 | 0.487±1.2e-3 | **0.552±2.7e-3** |
| | RotH (RotatE) | $\mathbb{H}$ ($\mathbb{E}$) | **0.474±2.3e-3** | **0.432±2.0e-3** | **0.491±3.1e-3** | 0.550±1.7e-3 |
| | RotatE (AttH) | $\mathbb{E}$ ($\mathbb{H}$) | 0.384±1.5e-3 | 0.354±1.2e-3 | 0.400±9.2e-4 | 0.433±9.8e-4 |
| | RotatE (RotH) | $\mathbb{E}$ ($\mathbb{H}$) | 0.384±2.2e-3 | 0.351±1.4e-3 | 0.403±1.8e-3 | 0.438±2.1e-3 |
| | TransE (AttH) | $\mathbb{E}$ ($\mathbb{H}$) | 0.245±1.1e-3 | 0.090±7.5e-4 | 0.366±9.8e-4 | 0.511±1.6e-3 |
| | TransE (RotH) | $\mathbb{E}$ ($\mathbb{H}$) | 0.247±1.3e-3 | 0.093±8.3e-4 | 0.370±1.1e-3 | 0.509±3.5e-3 |
| FB15K-237 | AttH (TransE) | $\mathbb{H}$ ($\mathbb{E}$) | 0.316±4.8e-4 | 0.225±3.6e-4 | **0.348±7.2e-4** | **0.500±5.5e-4** |
| | AttH (RotatE) | $\mathbb{H}$ ($\mathbb{E}$) | **0.318±8.6e-4** | **0.228±7.1e-4** | **0.348±9.3e-4** | 0.497±1.2e-3 |
| | RotH (TransE) | $\mathbb{H}$ ($\mathbb{E}$) | 0.313±1.1e-3 | 0.222±7.3e-4 | 0.344±1.3e-3 | 0.496±2.4e-3 |
| | RotH (RotatE) | $\mathbb{H}$ ($\mathbb{E}$) | 0.311±1.0e-3 | 0.222±6.7e-4 | 0.343±1.3e-3 | 0.488±2.7e-3 |
| | RotatE (AttH) | $\mathbb{E}$ ($\mathbb{H}$) | 0.298±2.4e-3 | 0.212±1.8e-3 | 0.326±1.6e-3 | 0.471±2.3e-3 |
| | RotatE (RotH) | $\mathbb{E}$ ($\mathbb{H}$) | 0.292±1.7e-3 | 0.207±9.2e-4 | 0.319±1.8e-3 | 0.463±1.05e-3 |
| | TransE (AttH) | $\mathbb{E}$ ($\mathbb{H}$) | 0.313±7.4e-4 | 0.221±1.0e-3 | 0.343±8.3e-4 | 0.495±9.9e-4 |
| | TransE (RotH) | $\mathbb{E}$ ($\mathbb{H}$) | 0.310±2.5e-3 | 0.220±1.9e-3 | 0.341±2.2e-3 | 0.492±3.1e-3 |

**Case Study 3** The previous experiments mainly evaluate the feasibility of fusing two KGE models, where these two models possess different relation-specific transformations and local/global views, and their the embedding spaces are identical. Thus, this case study mainly explores the fusing results of our framework in different embedding spaces and performs several experiments as follows. Here, we take Poincaré-based KGE models (AttH, RotH) and Euclidean-based KGE models (TransE and RotatE) as student model and auxiliary model respectively, and leverage the DMutDE framework to fuse their model views crossing different embedding spaces. And the experimental results are shown in Table VII. Compared with the vanilla KGE model in Table IV, there is no gains on the Poincaré-based KGE models, which means the different types of embedding space are hard to transfer the effective feature representations mutually. We believe that the difference of measure spaces determines the shape of the distributions of entity embeddings. For example, the Euclidean space is flattened and no-boundary, while the Poincaré space is embedded within a bounded Poincaré desk. Thus, in the process of the distillation, if the embeddings in Poincaré space comply with the distribution patterns of the embeddings in Euclidean space, then they will destroy geometric space constrains, leading to the degeneration of student model.

TABLE VIII
RESULTS OF THE DMUTDE ON 32 EMBEDDING DIMENSION FOR STUDENT MODEL AND 512 FOR AUXILIARY MODEL. "STU." IS STUDENT MODEL AND "AUX." IS AUXILIARY MODEL. COLUMN-WISE COMPARISON, FIRST BEST IS **BOLD**, SECOND BEST IS UNDERLINED. THE $a\,\mathrm{e}\,b = a \times 10^{b}$, FOR EXAMPLE, $1\,\mathrm{e}{-1} = 1 \times 10^{-1}$. THE METHODS WITH "*" INDICATES THE RESULTS OF THE METHODS ARE REFERENCED FROM ORIGINAL PAPER.

| Dataset | KGE Methods Stu.(Aux.) | BKD | | RKD | | TA | | MulDE* | | IterDE* | | DualDE* | | DMutDE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MRR | Hit@10 | MRR | Hit@10 | MRR | Hit@10 | MRR | Hit@10 | MRR | Hit@10 | MRR | Hit@10 | MRR | Hit@10 |
| WN18RR | TransE (TransE) | 0.235±1.5e-3 | 0.520±1.9e-3 | 0.263±1.2e-3 | 0.528±1.7e-3 | 0.253±8.2e-4 | 0.523±1.2e-3 | 0.209 | 0.499 | 0.218 | 0.505 | 0.210 | 0.484 | **0.268±1.1e-3** | 0.531±4.2e-3 |
| | TransE (RotatE) | 0.231±1.2e-3 | 0.517±2.0e-3 | 0.260±1.0e-3 | 0.524±2.1e-3 | 0.255±1.1e-3 | 0.528±9.3e-4 | - | - | - | - | - | - | **0.262±1.5e-3** | 0.525±6.5e-3 |
| | RotatE (RotatE) | 0.435±2.1e-3 | 0.489±2.3e-3 | 0.442±7.0e-4 | 0.503±1.1e-3 | 0.428±1.6e-3 | 0.470±3.2e-3 | 0.451 | 0.536 | **0.471** | 0.558 | 0.468 | 0.560 | 0.466±7.1e-4 | 0.565±3.2e-3 |
| | RotatE (TransE) | 0.441±9.4e-4 | 0.509±1.9e-3 | 0.439±9.3e-4 | 0.501±2.8e-3 | 0.437±3.1e-3 | 0.489±2.7e-3 | - | - | - | - | - | - | **0.468±1.1e-3** | 0.562±2.8e-3 |
| | AttH (AttH) | 0.455±1.2e-3 | 0.537±2.1e-3 | 0.461±1.1e-3 | 0.538±8.9e-4 | 0.447±2.2e-3 | 0.522±2.5e-3 | - | - | - | - | - | - | **0.465±6.6e-4** | 0.541±1.8e-3 |
| | AttH (RotH) | 0.443±1.7e-3 | 0.521±2.3e-3 | 0.452±9.4e-4 | 0.520±1.5e-3 | 0.450±1.8e-3 | 0.524±2.2e-3 | - | - | - | - | - | - | **0.468±3.6e-4** | 0.544±9.4e-4 |
| | AttH (RefH) | 0.456±2.1e-3 | 0.538±3.8e-3 | 0.453±2.3e-3 | 0.539±2.5e-3 | 0.450±1.6e-3 | 0.523±2.3e-3 | - | - | - | - | - | - | **0.460±8.9e-4** | 0.537±2.5e-3 |
| | RotH (RotH) | 0.474±9.9e-4 | 0.560±1.7e-3 | 0.473±1.2e-3 | 0.549±3.1e-3 | 0.468±2.8e-3 | 0.544±1.4e-3 | 0.481 | 0.574 | - | - | - | - | **0.482±1.1e-3** | 0.571±3.2e-3 |
| | RotH (RefH) | 0.469±7.8e-4 | 0.555±1.4e-3 | 0.463±8.9e-4 | 0.554±1.7e-3 | 0.470±2.1e-3 | 0.550±4.2e-3 | - | - | - | - | - | - | **0.477±7.2e-4** | 0.567±7.9e-4 |
| | RotH (AttH) | 0.471±1.0e-3 | 0.551±2.1e-3 | 0.464±1.2e-3 | 0.545±2.7e-3 | 0.470±2.5e-3 | 0.552±1.3e-3 | - | - | - | - | - | - | **0.483±1.3e-3** | 0.570±4.2e-3 |
| | RefH (RefH) | 0.464±5.2e-4 | 0.545±9.6e-4 | 0.458±6.1e-4 | 0.532±1.2e-3 | 0.460±5.5e-4 | 0.534±1.9e-3 | **0.479** | 0.569 | - | - | - | - | 0.478±5.3e-4 | 0.571±9.7e-4 |
| | RefH (AttH) | 0.463±1.0e-3 | 0.535±6.4e-4 | 0.459±8.2e-4 | 0.529±9.1e-4 | 0.462±1.2e-3 | 0.535±7.9e-4 | - | - | - | - | - | - | **0.480±2.3e-3** | 0.568±8.4e-3 |
| | RefH (RotH) | 0.462±6.2e-4 | 0.535±8.5e-4 | 0.458±7.3e-4 | 0.525±1.4e-3 | 0.460±1.1e-3 | 0.534±2.5e-3 | - | - | - | - | - | - | **0.479±4.7e-4** | 0.569±2.7e-3 |
| FB15K-237 | TransE (TransE) | 0.311±5.8e-4 | 0.490±9.1e-4 | 0.312±8.2e-4 | 0.493±1.0e-3 | 0.311±6.9e-4 | 0.489±2.1e-3 | 0.238 | 0.417 | 0.266 | 0.443 | 0.254 | 0.418 | **0.314±1.1e-3** | 0.496±1.8e-3 |
| | TransE (RotatE) | 0.307±6.6e-4 | 0.486±1.2e-3 | 0.306±9.0e-4 | 0.484±1.5e-3 | **0.311±8.2e-4** | 0.490±4.9e-4 | - | - | - | - | - | - | 0.310±7.3e-4 | 0.489±2.2e-3 |
| | RotatE (RotatE) | 0.273±1.1e-3 | 0.442±1.6e-3 | 0.279±8.9e-4 | 0.447±1.8e-3 | 0.277±5.8e-4 | 0.449±1.2e-3 | 0.300 | 0.477 | **0.310** | 0.492 | 0.306 | 0.487 | 0.305±7.2e-4 | 0.483±2.6e-3 |
| | RotatE (TransE) | 0.278±4.8e-4 | 0.447±7.7e-4 | 0.281±6.8e-4 | 0.450±1.2e-3 | 0.274±4.9e-4 | 0.442±9.1e-4 | - | - | - | - | - | - | **0.301±1.2e-3** | 0.479±9.2e-4 |
| | AttH (AttH) | 0.329±6.2e-4 | 0.508±1.1e-3 | 0.330±7.4e-4 | 0.512±8.9e-4 | 0.324±7.3e-4 | 0.509±1.1e-3 | - | - | - | - | - | - | **0.333±9.3e-4** | 0.517±5.9e-3 |
| | AttH (RotH) | 0.322±4.1e-4 | 0.502±8.3e-4 | 0.325±6.9e-4 | 0.506±4.8e-4 | 0.326±7.9e-4 | 0.505±1.0e-3 | - | - | - | - | - | - | **0.328±1.0e-3** | 0.513±3.9e-3 |
| | AttH (RefH) | 0.327±7.3e-4 | 0.509±1.2e-3 | 0.326±8.8e-4 | 0.507±9.2e-4 | 0.325±4.8e-4 | 0.508±1.4e-3 | - | - | - | - | - | - | **0.329±2.4e-3** | 0.511±8.8e-3 |
| | RotH (RotH) | 0.318±6.7e-4 | 0.496±1.3e-3 | 0.318±8.2e-4 | 0.497±1.5e-3 | 0.271±5.3e-4 | 0.423±9.9e-4 | 0.328 | 0.515 | - | - | - | - | **0.329±6.1e-4** | 0.518±9.2e-4 |
| | RotH (RefH) | 0.319±3.8e-4 | 0.501±8.4e-4 | 0.321±5.7e-4 | 0.503±1.1e-3 | 0.277±6.2e-4 | 0.432±9.3e-4 | - | - | - | - | - | - | **0.325±7.8e-4** | 0.510±3.9e-3 |
| | RotH (AttH) | 0.325±7.5e-4 | 0.506±9.2e-4 | 0.328±8.3e-4 | 0.507±1.8e-3 | 0.264±8.4e-4 | 0.415±1.2e-3 | - | - | - | - | - | - | **0.330±1.1e-3** | 0.521±4.2e-3 |
| | RefH (RefH) | 0.310±6.2e-4 | 0.487±1.0e-3 | 0.320±1.1e-3 | 0.496±8.3e-4 | 0.311±9.5e-4 | 0.487±2.0e-3 | **0.325** | 0.508 | - | - | - | - | 0.323±7.6e-4 | 0.509±3.6e-3 |
| | RefH (AttH) | 0.314±8.7e-4 | 0.489±1.4e-3 | 0.316±4.9e-4 | 0.495±1.6e-3 | 0.313±7.9e-4 | 0.490±1.7e-3 | - | - | - | - | - | - | **0.322±1.0e-3** | 0.507±2.8e-3 |
| | RefH (RotH) | 0.311±5.3e-4 | 0.485±9.0e-4 | 0.310±7.6e-4 | 0.485±9.7e-4 | 0.312±8.7e-4 | 0.487±1.2e-3 | - | - | - | - | - | - | **0.320±8.0e-4** | 0.500±4.6e-3 |

**Case Study 4** Can DMutDE still perform well on the setting that compresses the effective feature representations from the large teacher to the light-weighted student as what IterDE [64], DualDE [25] and MulDE [24] do? Here, we explore the performance of DMutDE on distilling a better 32-dimensional KGE model via a 512-dimensional teacher KGE model. The experimental results are shown in Table VIII. DMutDE achieves comparable performances comparing with other baselines, which proves the modules proposed in the paper are still effective for transferring the effective feature representations from large teacher model to small student models.

*3) **RQ3:** What is the contribution of each module?:* DMutDE framework mainly contains two modules, a soft label fusion module, and an entity embedding distillation module. We firstly discuss how much the DMutDE affects the training of these two peer KGE models compared with the vanilla KGE model training individually. Here we evaluate the training of AttH and RotH with/without DMutDE on FB15K-237. The Figure 7 shows the comparison of the training performance curves of AttH and RotH with/without DMutDE framework. It is obvious that the performances of these two models obtain significant improvements through the mutual distillation of DMutDE.

**Impacts of the Soft-Label Fusion and Entity Embedding Distillation.** To better understand the significance of the soft-label distillation and entity embedding distillation modules, we ablate the soft-label fusion module and entity embedding

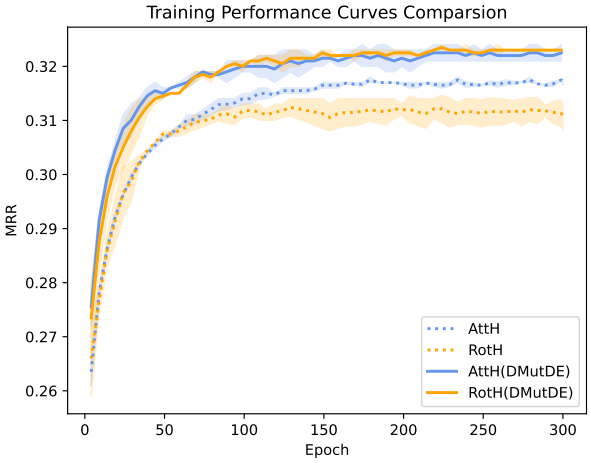


Fig. 7. Training performance curves of the AttH and RotH with/without the DMutDE.

TABLE IX
RESULTS OF ROTH IN DMUTDE, WHERE ROTH AS STUDENT ATTH AS AUXILIARY. THE TABLE SHOWS THE PERFORMANCE OF ROTH. ✗MEANS THE CORRESPONDING MODULE IS REMOVED. "SLF" IS SOFT LABEL FUSION, "EED" IS ENTITY EMBEDDING DISTILLATION.

| SLF | EED | WN18RR MRR | WN18RR Hit@10 | FB15K-237 MRR | FB15K-237 Hit@10 |
|---|---|---|---|---|---|
| ✗ | | 0.469±4.8e-4 | 0.553±1.7e-3 | 0.319±2.1e-4 | 0.503±6.4e-4 |
| | ✗ | 0.472±3.5e-4 | 0.538±1.3e-3 | 0.310±4.2e-4 | 0.484±8.3e-4 |

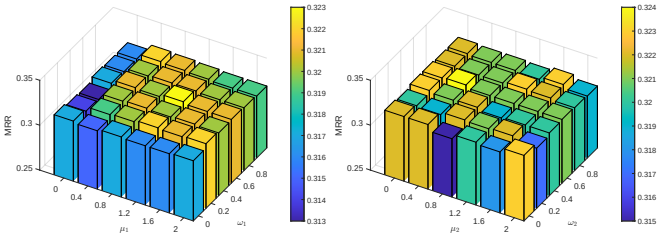


Fig. 8. Results of the impacts of soft-label fusion module and entity embedding distillation module. Varying $\mu_1, \mu_2$ in $[0, 2]$, $\omega_1, \omega_2$ in $[0, 1]$.

distillation module from DMutDE framework, respectively, to evaluate their contributions on WN18RR and FB15K-237 datasets. Specifically, we adopt RotH as student model and AttH as auxiliary model in DMutDE. We mainly report the relative experimental results of RotH in Table IX. Removing the entity embedding distillation module degrades distilled RotH performance to that of vanilla RotH, as it limits the increase in mutual information $I(z_{\mathrm{S}}; z_{\mathrm{A}})$ between the student and auxiliary models, which in turn restricts the mutual information $I(z_{\mathrm{S}}; \tilde{z})$ between the student model and the ideal KGE model. Furthermore, removing the soft-label fusion module results in uncontrolled information transfer and the propagation of heterogeneous noise during distillation, causing the student model to fully replicate the auxiliary model's prediction behavior.

Then, to study the effects of the hyperparameters $\omega_1$, $\mu_1$ of soft-label fusion module and the hyperparameters $\omega_2$, $\mu_2$ of entity embedding distillation module on the model's performance, we conduct the sensitivity analysis on the FB15K-237 dataset with DMutDE framework that employing AttH as auxiliary model and RotH as student model, as shown in Figure 8. Fixing the hyperparameters of Entity Embedding Distillation $\omega_2, \mu_2$, we vary the hyperparameters $\omega_1$ from $[0, 1]$ and $\mu_1$ from $[0, 2]$ of Soft-Label Fusion module. We find that the RotH achieves best performance when $\omega_1$ is 0.6 and $\mu_1$ is 1.2. This result highlights two key findings: 1. The fine-grained soft label distillation $L_{\mathrm{Soft}}^{\mathrm{Fine}}$ effectively extracts information from the distribution of negative sample scores. 2. The coarse-grained soft label distillation $L_{\mathrm{Soft}}^{\mathrm{Coarse}}$ suppresses the distribution of normalized negative sample scores due to excessively high positive sample scores, resulting in partial information loss. In the Entity Embedding Distillation module, the best hyperparameters $\omega_2$ is 0.6, $\mu_2$ is 0.4, which indicates the angle and length ratio are both important for embedding distillation.

**Entity Embedding Visualization.** To better demonstrate the effectiveness of the DMutDE framework, we visualize the learned entity embeddings of the vanilla KGE model and distilled the KGE model with our framework. We distill AttH and RotH mutually with our framework. And the FB15KET [79] dataset provides the entity type information for each entity. Then, we use t-SNE dimensional reduction algorithm [80] for the entity embedding visualization. Here we visualize the entity embeddings of AttH and RotH with/without distillation of DMutDE, and the visualization results are shown in Figure 9. The learned entity embeddings distilled with DMutDE can obtain better clustering performance than that of vanilla entity embeddings.

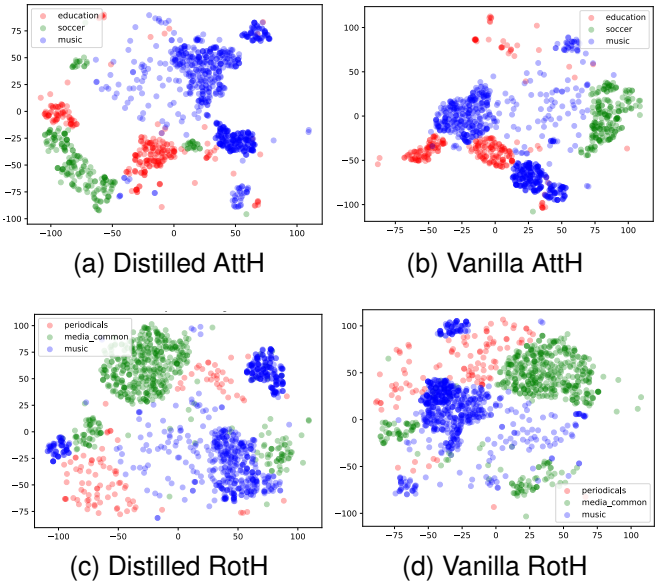


Fig. 9. The visualization of the learned entity embeddings distributions of the AttH and RotH. (a). The AttH is distilled with the DMutDE. (b). The vanilla AttH. (c). The RotH is distilled with the DMutDE. (d). The vanilla RotH.

## VII. Conclusion

In resource-limited scenarios, training large teacher models may be impractical. Instead, we enhance student performance by fusing model views from two distinct KGE methods. We propose a novel distillation framework, termed **D**ual-View **Mut**ual **D**istillation for Knowledge Graph **E**mbeddings (DMutDE). The discrepancy in relation mapping influences the model views, reflected in predicted entity rankings. Our framework transfers this knowledge using mutual distillation. Specifically, a soft label fusion module refines score distributions, and an entity embedding distillation module transfers structural features between models.

### A. Future Works

**Remaining Limitations.** We address a potential limitation in the experiment section, specifically in Case Study 3, where we explore mutual distillation across two different metric spaces: Poincaré and Euclidean. Future work will focus on mutual distillation between different KGE scoring functions. Current KGE scoring methods fall into two categories: semantic matching-based (using inner product) and spatial distance-based (using geodesic distance). In this paper, we focus on distillation between spatial distance-based models. Future research will extend to distillation between semantic matching-based models and across both scoring types. These discussions will included in future work.

**Potential Direction Beyond KGE.** This framework extends beyond KGE to continual learning. In our approach, two KGE

TABLE X
PARTIAL KEY HYPER-PARAMETERS OF DMUTDE ON WN18RR AND FB15K-237 DATASETS.

| Hyper-parameters | WN18RR | | | FB15K-237 | | |
|---|---|---|---|---|---|---|
| Student (Auxiliary) | RotH (AttH) | RotH (RotH) | RotH (RefH) | RotH (AttH) | RotH (RotH) | RotH (RefH) |
| #Neg Sampling | 50 | 50 | 50 | 100 | 100 | 100 |
| Optimizer | Adam | Adam | Adam | Adagrad | Adagrad | Adagrad |
| Embedding Rank | 32 | 32 | 32 | 32 | 32 | 32 |
| Batch Size | 8000 | 8000 | 8000 | 8000 | 8000 | 8000 |
| Dist Epoch $\theta$ | 60 | 60 | 60 | 40 | 40 | 40 |
| $\mu_1$ | 1.0 | 1.0 | 1.0 | 1.2 | 1.2 | 1.2 |
| $\omega_1$ | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 |
| $\mu_2$ | 0.2 | 0.2 | 0.2 | 0.1 | 0.1 | 0.1 |
| $\omega_2$ | 0.4 | 0.4 | 0.4 | 0.8 | 0.8 | 0.8 |

models exchange model views to enhance their performance. For model A, the view from model B is treated as new knowledge, analogous to new data, allowing for an extension to continual learning. Since incorporating new data can affect old data representations, a gating mechanism is needed to control the input of beneficial information and block harmful noise, similar to the $f_{\text{fused}}$ function in this paper. We adjust the quality assessment function to ensure positive samples score higher than negative ones, otherwise treating them as noise. In continual learning, a similar function is defined to evaluate the quality of newly injected information, distinguishing noise from valid input.

## REFERENCES


[1] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 697–706. [Online]. Available: https://doi.org/10.1145/1242572.1242667

[2] F. Mahdisoltani, J. A. Biega, and F. M. Suchanek, "Yago3: A knowledge base from multilingual wikipedias," in *Conference on Innovative Data Systems Research*, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:6611164

[3] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "Dbpedia - a crystallization point for the web of data," *J. Web Semant.*, vol. 7, pp. 154–165, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:16081721

[4] K. Luo, F. Lin, X. Luo, and K. Zhu, "Knowledge base question answering via encoding of complex query graphs," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2185–2194.

[5] S. Lv, D. Guo, J. Xu, D. Tang, N. Duan, M. Gong, L. Shou, D. Jiang, G. Cao, and S. Hu, "Graph-based reasoning over heterogeneous external knowledge for commonsense question answering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 8449–8456.

[6] H. Terdalkar and A. Bhattacharya, "Framework for question-answering in sanskrit through automated construction of knowledge graphs," in *Proceedings of the 6th International Sanskrit Computational Linguistics Symposium*, 2019, pp. 97–116.

[7] M. Welling and T. N. Kipf, "Semi-supervised classification with graph convolutional networks," in *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

[8] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 353–362.

[9] M. Gaur, K. Gunaratna, V. Srinivasan, and H. Jin, "Iseeq: Information seeking question generation using dynamic meta-information retrieval and knowledge graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 10 672–10 680.

[10] C. Wise, V. N. Ioannidis, M. R. Calvo, X. Song, G. Price, N. Kulkarni, R. Brand, P. Bhatia, and G. Karypis, "Covid-19 knowledge graph: accelerating information retrieval and discovery for scientific literature," *arXiv preprint arXiv:2007.12731*, 2020.

[11] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong and M. Strube, Eds. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 687–696. [Online]. Available: https://aclanthology.org/P15-1067

[12] R. Xie, Z. Liu, and M. Sun, "Representation learning of knowledge graphs with hierarchical types," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI'16. AAAI Press, 2016, p. 2965–2971.

[13] W. Zheng, L. Yin, X. Chen, Z. Ma, S. Liu, and B. Yang, "Knowledge base graph embedding module design for visual question answering model," *Pattern Recognition*, vol. 120, p. 108153, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003132032100340X

[14] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[15] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.

[16] S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion knowledge graph embeddings," *Advances in neural information processing systems*, vol. 32, 2019.

[17] I. Balazevic, C. Allen, and T. Hospedales, "Multi-relational poincaré graph embeddings," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[18] I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi, and C. Ré, "Low-dimensional hyperbolic knowledge graph embeddings," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6901–6914.

[19] M. Nayyeri, C. Xu, F. Hoffmann, M. M. Alam, J. Lehmann, and S. Vahdati, "Knowledge graph representation learning using ordinary differential equations," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 9529–9548.

[20] I. Balazevic, C. Allen, and T. Hospedales, "TuckER: Tensor factorization for knowledge graph completion," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5185–5194. [Online]. Available: https://aclanthology.org/D19-1522

[21] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.

[22] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[23] K. Xu, D. H. Park, C. Yi, and C. Sutton, "Interpreting deep classifier by visual distillation of dark knowledge," *arXiv preprint arXiv:1803.04042*, 2018.

[24] K. Wang, Y. Liu, Q. Ma, and Q. Z. Sheng, "Mulde: Multi-teacher knowledge distillation for low-dimensional knowledge graph embeddings," in *Proceedings of the Web Conference 2021*, 2021, pp. 1716–1726.

[25] Y. Zhu, W. Zhang, M. Chen, H. Chen, X. Cheng, W. Zhang, and H. Chen, "Dualde: Dually distilling knowledge graph embedding for faster and cheaper reasoning," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 1516–1524.

[26] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 2, pp. 494–514, 2021.

[27] J. Cao, J. Fang, Z. Meng, and S. Liang, "Knowledge graph embedding: A survey from the perspective of representation spaces," *arXiv preprint arXiv:2211.03536*, 2022.

[28] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Icml*, 2011.

[29] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *arXiv preprint arXiv:1412.6575*, 2014.

[30] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, no. 1, 2014.

[31] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[32] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 2015, pp. 687–696.

[33] J. Guo and S. Kok, "Bique: Biquaternionic embeddings of knowledge graphs," *arXiv preprint arXiv:2109.14401*, 2021.

[34] C. Gao, C. Sun, L. Shan, L. Lin, and M. Wang, "Rotate3d: Representing relations as rotations in three-dimensional space for knowledge graph embedding," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 385–394.

[35] Y. Tang, J. Huang, G. Wang, X. He, and B. Zhou, "Orthogonal relation transforms with graph context modeling for knowledge graph embedding," *arXiv preprint arXiv:1911.04910*, 2019.

[36] T. Ebisu and R. Ichise, "Toruse: Knowledge graph embedding on a lie group," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[37] H. Yang and J. Liu, "Knowledge graph representation learning as groupoid: unifying transe, rotate, quate, complex," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 2311–2320.

[38] H. Xiao, M. Huang, and X. Zhu, "From one point to a manifold: Knowledge graph embedding for precise link prediction," *arXiv preprint arXiv:1512.04792*, 2015.

[39] J. W. Cannon, W. J. Floyd, R. Kenyon, W. R. Parry *et al.*, "Hyperbolic geometry," *Flavors of geometry*, vol. 31, no. 59-115, p. 2, 1997.

[40] O. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," *Advances in neural information processing systems*, vol. 31, 2018.

[41] I. Chami, Z. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," *Advances in neural information processing systems*, vol. 32, 2019.

[42] Z. Pan and P. Wang, "Hyperbolic hierarchy-aware knowledge graph embedding for link prediction," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 2941–2948.

[43] Z. Sun, M. Chen, W. Hu, C. Wang, J. Dai, and W. Zhang, "Knowledge association with hyperbolic knowledge graph embeddings," *arXiv preprint arXiv:2010.02162*, 2020.

[44] A. Gu, F. Sala, B. Gunel, and C. Ré, "Learning mixed-curvature representations in product spaces," in *International Conference on Learning Representations*, 2018.

[45] B. Xiong, S. Zhu, M. Nayyeri, C. Xu, S. Pan, C. Zhou, and S. Staab, "Ultrahyperbolic knowledge graph embeddings," *arXiv preprint arXiv:2206.00449*, 2022.

[46] M. Law and J. Stam, "Ultrahyperbolic representation learning," *Advances in neural information processing systems*, vol. 33, pp. 1668–1678, 2020.

[47] Z. Cao, Q. Xu, Z. Yang, X. Cao, and Q. Huang, "Geometry interaction knowledge graph embeddings," in *AAAI Conference on Artificial Intelligence*, 2022.

[48] M. Nayyeri, C. Xu, F. Hoffmann, M. M. Alam, J. Lehmann, and S. Vahdati, "Knowledge graph representation learning using ordinary differential equations," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 9529–9548.

[49] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

[50] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, M. Walker, H. Ji, and A. Stent, Eds. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 327–333. [Online]. Available: https://aclanthology.org/N18-2053

[51] Z. Li, H. Liu, Z. Zhang, T. Liu, and N. N. Xiong, "Learning knowledge graph embedding with heterogeneous relation attention networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3961–3973, 2022.

[52] Q. Li, D. Wang, S. Feng, C. Niu, and Y. Zhang, "Global graph attention embedding network for relation prediction in knowledge graphs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 11, pp. 6712–6725, 2022.

[53] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*. Springer, 2018, pp. 593–607.

[54] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 950–958.

[55] Y. Zhang, Q. Yao, W. Dai, and L. Chen, "Autosf: Searching scoring functions for knowledge graph embedding," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 2020, pp. 433–444.

[56] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

[57] D. Jiang, R. Wang, L. Xue, and J. Yang, "Multisource hierarchical neural network for knowledge graph embedding," *Expert Syst. Appl.*, vol. 237, no. PB, feb 2024. [Online]. Available: https://doi.org/10.1016/j.eswa.2023.121446

[58] T. Le, H. Tran, and B. Le, "Knowledge graph embedding with the special orthogonal group in quaternion space for link prediction," *Knowledge-Based Systems*, vol. 266, p. 110400, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705123001508

[59] J. Lee, C. Chung, and J. J. Whang, "Ingram: inductive knowledge graph embedding via relation graphs," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[60] D. Xu, Z. Zhang, Z. Lin, X. Wu, Z. Zhu, T. Xu, X. Zhao, Y. Zheng, and E. Chen, "Multi-perspective improvement of knowledge graph completion with large language models," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, Eds. Torino, Italia: ELRA and ICCL, May 2024, pp. 11 956–11 968. [Online]. Available: https://aclanthology.org/2024.lrec-main.1044

[61] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3967–3976.

[62] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 5191–5198.

[63] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9163–9171.

[64] J. Liu, P. Wang, Z. Shang, and C. Wu, "Iterde: an iterative knowledge distillation framework for knowledge graph embeddings," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 4488–4496.

[65] X. Zhu, G. Li, and W. Hu, "Heterogeneous federated knowledge graph embedding learning and unlearning," in *Proceedings of the ACM Web Conference 2023*, ser. WWW '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 2444–2454. [Online]. Available: https://doi.org/10.1145/3543507.3583305

[66] J. Liu, W. Ke, P. Wang, Z. Shang, J. Gao, G. Li, K. Ji, and Y. Liu, "Towards continual knowledge graph embedding via incremental distillation," in *AAAI Conference on Artificial Intelligence*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:268692961

[67] W. Webber, A. Moffat, and J. Zobel, "A similarity measure for indefinite rankings," *ACM Trans. Inf. Syst.*, vol. 28, no. 4, nov 2010. [Online]. Available: https://doi.org/10.1145/1852102.1852106

[68] E. Beltrami, "Teoria fondamentale degli spazii di curvatura costante," *Annali di Matematica Pura ed Applicata (1867-1897)*, vol. 2, no. 1, pp. 232–255, 1868.

[69] W. Peng, T. Varanka, A. Mostafa, H. Shi, and G. Zhao, "Hyperbolic deep neural networks: A survey," *arXiv preprint arXiv:2101.04562*, 2021.

[70] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1–1, 01 2021.

[71] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/ea8fcd92d59581717e06eb187f10666d-Paper.pdf

[72] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *CoRR*, vol. abs/1412.6550, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:2723173

[73] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4320–4328, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:26071966

[74] K. Gokcesu and H. Gokcesu, "Generalized huber loss for robust learning and its efficient minimization for a robust statistics," *ArXiv*, vol. abs/2108.12627, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:237353039

[75] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, "Representing text for joint embedding of text and knowledge bases," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, L. Màrquez, C. Callison-Burch, and J. Su, Eds. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1499–1509. [Online]. Available: https://aclanthology.org/D15-1174

[76] G. A. Miller, "Wordnet: a lexical database for english," *Commun. ACM*, vol. 38, no. 11, p. 39–41, nov 1995. [Online]. Available: https://doi.org/10.1145/219717.219748

[77] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 1247–1250. [Online]. Available: https://doi.org/10.1145/1376616.1376746

[78] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[79] C. Moon, P. Jones, and N. F. Samatova, "Learning entity type embeddings for knowledge graph completion," in *Proceedings of the 2017 ACM on conference on information and knowledge management*, 2017, pp. 2215–2218.

[80] T. T. Cai and R. Ma, "Theoretical foundations of t-sne for visualizing high-dimensional clustered data," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 13 581–13 634, 2022.