

Duality-Induced Regularizer for Semantic Matching Knowledge Graph Embeddings

Jie Wang, *Senior Member, IEEE*, Zhanqiu Zhang, Zhihao Shi, Jianyu Cai,
Shuiwang Ji, *Senior Member, IEEE*, and Feng Wu, *Fellow, IEEE*

Abstract—Semantic matching models—which assume that entities with similar semantics have similar embeddings—have shown great power in knowledge graph embeddings (KGE). Many existing semantic matching models use inner products in embedding spaces to measure the plausibility of triples and quadruples in static and temporal knowledge graphs. However, vectors that have the same inner products with another vector can still be orthogonal to each other, which implies that entities with similar semantics may have dissimilar embeddings. This property of inner products significantly limits the performance of semantic matching models. To address this challenge, we propose a novel regularizer—namely, **D****U**ality-**I**nduced **R**egul**A**rizer (DURA)—which effectively encourages the entities with similar semantics to have similar embeddings. The major novelty of DURA is based on the observation that, for an existing semantic matching KGE model (*primal*), there is often another distance based KGE model (*dual*) closely associated with it, which can be used as effective constraints for entity embeddings. Experiments demonstrate that DURA consistently and significantly improves the performance of state-of-the-art semantic matching models on both static and temporal knowledge graph benchmarks.

Index Terms—Knowledge Graph Embeddings, Knowledge Graph, Link Prediction, Temporal Knowledge Graphs, Regularization.

1 INTRODUCTION

KNOWLEDGE graphs store human knowledge in a structured way, with nodes being entities and edges being relations. In the past few years, knowledge graphs have made great achievements in many areas, such as natural language processing [1], question answering [2], recommendation systems [3], and computer vision [4].

Some popular knowledge graphs, such as Freebase [5], Wikidata [6], and Yago3 [7], consists of triples (subject, predicate, object). However, in many scenarios, triples cannot well represent knowledge as knowledge may evolve with time. For example, the fact (Donald Trump, president, USA) is only valid at the time interval 2017-2021. Therefore, temporal knowledge graphs that consists of quadruples (subject, predicate, object, timestamp) also attract increasing attention recently. Although commonly used knowledge graphs usually contain billions of triples or quadruples, they still suffer from the incompleteness problem that a lot of factual triples or quadruples are missing. Due to the large scale of knowledge graphs, it is impractical to find all valid triples or quadruples manually. Therefore, knowledge graph completion (KGC)—which aims to predict missing links between entities based on known links automatically—has attracted much attention.

Knowledge graph embedding (KGE) is a powerful technique for KGC, which embeds entities, relations, and possible timestamps in KGs into low-dimensional continu-

ous embedding spaces. KGE models define a score function for each triple/quadruple in embedding spaces. Valid triples/quadruples are expected to have higher scores than invalid ones. The success of many existing KGE models relies on an assumption that entities with similar semantics should have similar representations. For example, the triples (`lions`, `is`, `mammals`) and (`tigers`, `is`, `mammals`) will have similar scores if the entities `lions` and `tiger` have similar embeddings. In other words, given a known valid triple (`lions`, `is`, `mammals`), we can infer that the triple (`tigers`, `is`, `mammals`) is also valid with high confidence.

Semantic matching (SM) models are an important suite of knowledge graph embedding approaches. They usually model static and temporal knowledge graphs as partially observed third-order and fourth-order binary tensors, respectively. Then, they formulate KGC as a tensor completion problem and try to solve it by tensor factorization. From the perspective of scoring function, SM models use inner products between embeddings to define the plausibility of given triples and quadruples. Theoretically, these models are highly expressive and can well handle complex relation patterns, such as 1-N, N-1, and N-N relations [8], [9]. However, due to the property of inner products, entities with similar semantics may have dissimilar embeddings. For example, even if the embeddings of `lions` and `tiger` are orthogonal, the triples (`lions`, `is`, `mammals`) and (`tigers`, `is`, `mammals`) are likely to have the same scores when the two entity embeddings have proper norms (please refer to Section 4.1 for more detailed explanations). Similarly, we can observe the phenomenon in temporal semantic matching models (please refer to Section 5.3). The performance of SM models suffer from this phenomenon and thus cannot achieve state-of-the-art.

To tackle the challenge, we propose a novel regularizer for semantic matching KGE models—namely, **D****U**ality-**I**nduced **R**egul**A**rizer (DURA).

• J. Wang, Z. Zhang, Z. Shi, J. Cai, F. Wu are with: a) CAS Key Laboratory of Technology in GIPAS, University of Science and Technology of China, Hefei 230027, China; b) Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230091, China. E-mail: jiewangx@ustc.edu.cn, zqzhang@mail.ustc.edu.cn, zhihaoshi@mail.ustc.edu.cn, jycai@mail.ustc.edu.cn, fengwu@ustc.edu.cn.
• S. Ji is with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX77843 USA. E-mail: sji@tamu.edu.

induced **RegulArizer** (DURA)—which effectively encourages entities with similar semantics to have similar embeddings. Specifically, by noticing that entities with similar semantics often connect to the same entity through the same relation, we propose to use a regularizer to constrain these entities’ embeddings. Further, we propose DURA based on the observation called *duality*—for an existing semantic matching KGE model (*primal*), there is often another distance based KGE model closely associated with it (*dual*), which uses Minkowski distances to define score functions. The duality can be derived by expanding the squared score functions of the associated distance based models, where the cross-term in the expansion is exactly a SM model and the squared terms in it give us a regularizer. In other words, adding DURA to a SM model is equivalently to a SM model with its associative distance based model being regularization. Since the minimization of the Minkowski distances between vectors force these vectors to be the same, DURA indeed encourages entities with similar semantics to have similar embeddings. Moreover, we show that when relation embeddings are diagonal matrices, the minimization of DURA has a form of the tensor nuclear 2-norm [10]. Experiments demonstrate that DURA is widely applicable to various static and temporal semantic matching KGE models and yields consistent and significant improvements on benchmark datasets.

An earlier version of this paper has been published at NeurIPS 2020 [11]. This journal manuscript significantly extends the initial version in several aspects. First, we propose a temporal version of DURA, named TDURA, which is designed for temporal knowledge graph embeddings. Second, we introduce a general semantic matching temporal KGE model, named TRESCAL, which is an extension of RESCAL to temporal KGs. Third, we rigorously analyze the diagonal relation embedding matrices and show that the minimization of TDURA also has a form of the tensor nuclear 2-norm [10]. Finally, we conduct experiments on temporal knowledge graphs and conduct more ablation studies to demonstrate the effectiveness of TDURA.

2 PRELIMINARIES

We review the backgrounds of the knowledge graph, knowledge graph completion, and knowledge graph embeddings in Sections 2.1, 2.2, and 2.3, respectively. Then, we introduce the notations in Section 2.4.

2.1 Knowledge Graphs

Static Knowledge Graphs Given a set \mathcal{E} of entities and a set \mathcal{R} of relations, a knowledge graph $\mathcal{K} = \{(u_i, r_j, v_k)\} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is a set of triplets, where u_i and r_j are the i -th entity and j -th relation, respectively.

Temporal Knowledge Graphs Given a set \mathcal{E} of entities, a set \mathcal{R} of relations, and a set \mathcal{T} of timestamp, a temporal knowledge graph $\mathcal{K}_T = \{(u_i, r_j, v_k, t_l)\} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T}$ is a set of quadruples, where u_i, r_j and t_l are the i -th entity, j -th relation and l -th timestamp, respectively.

2.2 Knowledge Graph Completion

Static Knowledge Graph Completion (KGC) The goal of KGC is to predict valid but unobserved triples based on the

known triples in \mathcal{K} , which is often solved by knowledge graph embedding (KGE) models. KGE models associate each entity $u_i \in \mathcal{E}$ and relation $r_j \in \mathcal{R}$ with an embedding (may be real or complex vectors, matrices, and tensors) \mathbf{u}_i and \mathbf{r}_j . Generally, they define a score function $s : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R}$ to associate a score $s(u_i, r_j, v_k)$ with each potential triplet $(u_i, r_j, v_k) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. The scores measure the plausibility of triples. For a query $(u_i, r_j, ?)$, KGE models first fill the blank with each entity in the knowledge graphs and then score the resulted triples. Valid triples are expected to have higher scores than invalid triples.

Temporal Knowledge Graph Completion (TKGC) The goal of TKGC is to predict valid but unobserved quadruples based on the known quadruples in \mathcal{K}_T . TKGE models also contain two important categories: distance based models and semantic matching models, both of which are temporal knowledge graph embedding (TKGE) methods. TKGE models also associate each entity $u_i \in \mathcal{E}$, relation $r_j \in \mathcal{R}$, and timestamp $t_l \in \mathcal{T}$ with an embedding (may be real or complex vectors, matrices, and tensors) $\mathbf{u}_i, \mathbf{r}_j$ and \mathbf{t}_l . Generally, they define a score function $s : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T} \rightarrow \mathbb{R}$ to associate a score $s(u_i, r_j, v_k, t_l)$ with each potential quadruples $(u_i, r_j, v_k, t_l) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T}$. The scores measure the plausibility of quadruples. For a query $(u_i, r_j, ?, t_l)$, TKGE models first fill the blank with each entity in the temporal knowledge graphs and then score the resulted quadruples. Valid quadruples are expected to have higher scores than invalid quadruples.

2.3 Knowledge Graphs Embeddings

Knowledge graph embeddings aim to embed entities, relations, and possible timestamps in static and temporal knowledge graphs into low-dimensional continuous vector spaces, which have two important categories—distance based models and semantic matching models.

Distance Based (DB) Models for Static KGs DB models define the score function s with Minkowski distances. That is, the score functions have the formulation of $s(u_i, r_j, v_k) = -\|\Gamma(u_i, r_j, v_k)\|_p$, where Γ is a model-specific function.

Semantic Matching (SM) Models for Static KGs SM models regard a knowledge graph as a third-order binary tensor $\mathcal{X} \in \{0, 1\}^{|\mathcal{E}| \times |\mathcal{R}| \times |\mathcal{E}|}$. The (i, j, k) entry $\mathcal{X}_{ijk} = 1$ if (u_i, r_j, v_k) is valid otherwise $\mathcal{X}_{ijk} = 0$. Suppose that \mathcal{X}_j denotes the j -th frontal slice of \mathcal{X} , i.e., the adjacency matrix of the j -th relation. Generally, a SM KGE model factorizes \mathcal{X}_j as $\mathcal{X}_j \approx \text{Re}(\bar{\mathbf{U}}\mathbf{R}_j\mathbf{V}^\top)$, where the i -th (k -th) row of \mathbf{U} (\mathbf{V}) is \mathbf{u}_i (\mathbf{v}_k), \mathbf{R}_j is a matrix representing relation r_j , $\text{Re}(\cdot)$ and $\bar{\cdot}$ are the real part and the conjugate of a complex matrix, respectively. That is, the score functions are defined as $s(u_i, r_j, v_k) = \text{Re}(\bar{\mathbf{u}}_i\mathbf{R}_j\mathbf{v}_k^\top)$. Note that the real part and the conjugate of a real matrix are itself. The aim of SM models is to seek matrices $\mathbf{U}, \mathbf{R}_1, \dots, \mathbf{R}_{|\mathcal{R}|}, \mathbf{V}$, such that $\text{Re}(\bar{\mathbf{U}}\mathbf{R}_j\mathbf{V}^\top)$ can approximate \mathcal{X}_j . Let $\hat{\mathcal{X}}_j = \text{Re}(\bar{\mathbf{U}}\mathbf{R}_j\mathbf{V}^\top)$ and $\hat{\mathcal{X}}$ be a tensor of which the j -th frontal slice is $\hat{\mathcal{X}}_j$. Then the regularized formulation of a semantic matching model can be written as

$$\min_{\hat{\mathcal{X}}_1, \dots, \hat{\mathcal{X}}_{|\mathcal{R}|}} \sum_{j=1}^{|\mathcal{R}|} L(\mathcal{X}_j, \hat{\mathcal{X}}_j) + \lambda g(\hat{\mathcal{X}}), \quad (1)$$

where $L(\mathcal{X}_j, \hat{\mathcal{X}}_j)$ measures the discrepancy between \mathcal{X}_j , $\hat{\mathcal{X}}_j$, and g is the regularization function, and $\lambda > 0$ is a fixed parameter.

Distance Based (DB) Models for Temporal KGs Temporal DB models define the score function s with the Minkowski distance. That is, the score functions have the formulation of $s(u_i, r_j, v_k, t_l) = -\|\Gamma(u_i, r_j, v_k, t_l)\|_p$, where Γ is a model-specific function.

Semantic Matching (SM) Models for Temporal KGs Temporal SM models regard a temporal knowledge graph as a fourth-order binary tensor $\mathcal{X} \in \{0, 1\}^{|\mathcal{E}| \times |\mathcal{R}| \times |\mathcal{E}| \times |\mathcal{T}|}$. The (i, j, k, l) entry $\mathcal{X}_{ijkl} = 1$ if (u_i, r_j, v_k, t_l) is valid otherwise $\mathcal{X}_{ijkl} = 0$. Suppose that \mathcal{X}_{jl} denotes the (j, l) slice of \mathcal{X} , i.e., the adjacency matrix given the j -th relation and the l -th timestamp. Generally, a SM temporal KGE model factorizes \mathcal{X}_{jl} as $\mathcal{X}_{jl} \approx \text{Re}(\bar{\mathbf{U}}(\mathbf{R}_j \odot \mathbf{T}_l)\mathbf{V}^\top)$, where the i -th (k -th) row of \mathbf{U} (\mathbf{V}) is \mathbf{u}_i (\mathbf{v}_k), \mathbf{R}_j is a matrix representing relation r_j , \mathbf{T}_l is a matrix representing timestamp t_l , $\text{Re}(\cdot)$ and $\bar{\cdot}$ are the real part and the conjugate of a complex matrix, and \odot is the element-wise multiplication between two matrices. The aim of temporal SM models is to seek matrices $\mathbf{U}, \mathbf{R}_1, \dots, \mathbf{R}_{|\mathcal{R}|}, \mathbf{V}$ and $\mathbf{T}_1, \dots, \mathbf{T}_{|\mathcal{T}|}$, such that $\text{Re}(\bar{\mathbf{U}}(\mathbf{R}_j \odot \mathbf{T}_l)\mathbf{V}^\top)$ can approximate \mathcal{X}_{jl} . Let $\hat{\mathcal{X}}_{jl} = \text{Re}(\bar{\mathbf{U}}(\mathbf{R}_j \odot \mathbf{T}_l)\mathbf{V}^\top)$ and $\hat{\mathcal{X}}$ be a tensor of which the (j, l) slice is $\hat{\mathcal{X}}_{jl}$. Then the regularized formulation of a semantic matching model can be written as

$$\min_{\hat{\mathcal{X}}_{jl}} \sum_{j=1}^{|\mathcal{R}|} \sum_{l=1}^{|\mathcal{T}|} L(\mathcal{X}_{jl}, \hat{\mathcal{X}}_{jl}) + \lambda g(\hat{\mathcal{X}}), \quad (2)$$

where $\lambda > 0$ is a fixed parameter, $L(\mathcal{X}_{jl}, \hat{\mathcal{X}}_{jl})$ measures the discrepancy between \mathcal{X}_{jl} and $\hat{\mathcal{X}}_{jl}$, and g is the regularization function.

2.4 Other Notations

We use $u_i \in \mathcal{E}$ and $v_k \in \mathcal{E}$ to distinguish head and tail entities. Let $\|\cdot\|_1$, $\|\cdot\|_2$, and $\|\cdot\|_F$ denote the L_1 norm, the L_2 norm, and the Frobenius norm of matrices or vectors. We use $\langle \cdot, \cdot \rangle$ to represent the inner products of two real or complex vectors.

3 RELATED WORK

This work is related to knowledge graph embedding models and regularizer for semantic matching knowledge graph embeddings.

3.1 Knowledge Graph Embedding Models

Popular knowledge graph embeddings include distance-based, learning-based, and semantic matching models.

Distance based models describe relations as relational maps between head and tail entities. Then, they use the Minkowski distance to measure the plausibility of a given triplet or quadruple. Lower distances correspond to higher plausibility of given triplets or quadruples. For example, TransE [12] and its variants [8], [9] represent relations as translations in vector spaces. They assume that a valid triplet (u_i, r_j, v_k) satisfies $\mathbf{u}_{i,r_j} + \mathbf{r}_j \approx \mathbf{v}_{k,r_j}$, where \mathbf{u}_{i,r_j} and \mathbf{v}_{k,r_j} mean that entity embeddings may be relation-specific. Structured embedding (SE) [13] uses linear maps

to represent relations. Its score function is defined as $s(u_i, r_j, v_k) = -\|\mathbf{R}_j^1 \mathbf{u}_i - \mathbf{R}_j^2 \mathbf{v}_k\|_1$. RotatE [14] defines each relation as a rotation in a complex vector space and the score function is defined as $s(u_i, r_j, v_k) = -\|\mathbf{u}_i \circ \mathbf{r}_j - \mathbf{v}_k\|_1$, where $\mathbf{u}_i, \mathbf{r}_j, \mathbf{v}_k \in \mathbb{C}^k$ and $\|\mathbf{r}_i\| = 1$. Mode [15] assumes that \mathbf{R}_j^1 is diagonal and \mathbf{R}_j^2 is an identity matrix. It shares a similar score function $s(u_i, r_j, v_k) = -\|\mathbf{u}_i \circ \mathbf{r}_j - \mathbf{v}_k\|_1$ with RotatE but $\mathbf{u}_i, \mathbf{r}_j, \mathbf{v}_k \in \mathbb{R}^k$. Some works extend existing distance-based models to temporal KGs. For example, TTransE [16] extends TransE by learning timestamp embeddings. Inspired by TransH [8], HyTE [17] projects entity and relation embeddings into the timestamp-specific hyperplanes.

Learning-based models use neural networks—such as convolutional neural networks [18], capsule networks [19], and graph neural networks [20]—to model the interaction among triplets or quadruples. Then, they use either inner products or distance functions to measure the plausibility of given triplets or quadruples.

Semantic matching models usually formulate the KGC task as a third-order binary tensor completion problem. RESCAL [21] factorizes the j -th frontal slice of \mathcal{X} as $\mathcal{X}_j \approx \mathbf{A}\mathbf{R}_j\mathbf{A}^\top$. That is, embeddings of head and tail entities are from the same space. As the relation specific matrices contain lots of parameters, RESCAL is prone to be overfitting. DistMult [22] simplifies the matrix \mathbf{R}_j in RESCAL to be diagonal, while it sacrifices the expressiveness of models and can only handle symmetric relations. In order to model asymmetric relations, ComplEx [23] extends DistMult to complex embeddings. Both DistMult and ComplEx can be regarded as variants of canonical decomposition (CP) [24], which are in real and complex vector spaces, respectively. Some works also extend semantic matching models to temporal KGs. For example, ConT [25] extends TuckER [26] by additionally learning timestamp embeddings. ASALSAN [27] use three-way DEDICOM [28] to express temporal relations, which shares similar ideas with RESCAL [21]. Besides, TComplEx and TNTComplEx [29] reduce memory complexity by extending ComplEx to decomposition of the 4-order tensor.

Compared with distance-based models, semantic matching knowledge graph embeddings are more time-efficient, as we can easily parallel the computation of linear transformations and inner products. Moreover, due to the property of inner products, semantic matching is good at modeling one-to-many/many-to-one/many-to-many relations. For example, given the entity embedding \mathbf{u} and relation embedding \mathbf{r} , we can find different entity embeddings \mathbf{v}_i such that (u, r, v_i) have the same scores. In order to handle complex relations, distance-based models usually turn to relation-specific projection and further increase the computational burden. Compared with learning-based model, semantic matching knowledge graph embeddings are more time-efficient thanks to their simpler formulations.

However, also due to the property of inner products, in semantic matching, entities with similar semantics may have dissimilar embeddings, which will degrade the knowledge graph completion performance. As introduced in Section 4.1, even if the embeddings of Laue and Schottky are orthogonal, the triples $(\text{Laue}, \text{StudiedIn}, \text{HU Berlin})$ and $(\text{Schottky}, \text{StudiedIn}, \text{HU Berlin})$ are likely to have the same scores when the two entity embeddings have

proper norms. Moreover, distance-based models are better at modeling properties of knowledge graphs than semantic matching. For example, RotatE [14] can handle composition of relations; HAKE [15] and MuRP [30] can model semantic hierarchies.

It is worth noting that many general embedding learning methods are closely related to knowledge graph embeddings, all of which aim to optimize similarity between objects. For example, some graph embedding models [31], [32], [33] define similarity functions between nodes, and more generally, some feature embedding approaches [34], [35], [36] learn discriminative feature representations by optimizing similarity between samples. Different from these works, knowledge graph embeddings need to model complex relations between entities, leading to a more challenging problem.

3.2 Regularizers for Semantic Matching Knowledge Graph Embeddings

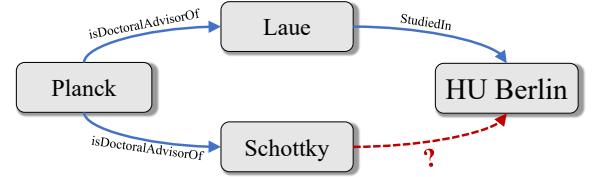
Semantic matching (SM) KGE models usually suffer from overfitting problem seriously, which motivates various regularizers. In the original papers of SM models, the authors usually use the squared Frobenius norm (L_2 norm) regularizer [21], [22], [23]. This regularizer cannot bring satisfying improvements. Consequently, SM models do not gain comparable performance to distance based models [14], [15]. More recently, [37] propose to use the tensor nuclear 3-norm [10] (N3) as a regularizer, which brings more significant improvements than the squared Frobenius norm regularizer. However, it is designed for the CP-like models, such as CP and ComplEx, and not suitable for more general models such as RESCAL. Moreover, some regularization methods aim to leverage external background knowledge. For example, to model equivalence and inversion axioms, [38] impose a set of model-dependent soft constraints on the predicate embeddings. [39] use non-negativity constraints on entity embeddings and approximate entailment constraints on relation embeddings to impose prior beliefs upon the structure of the embeddings space.

4 DURA FOR STATIC KGE

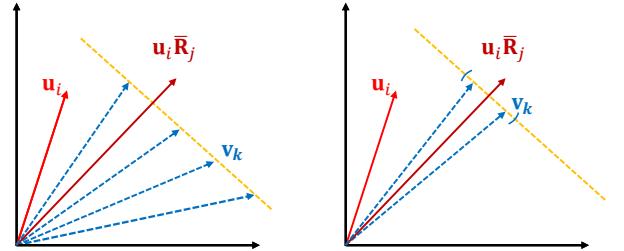
We introduce a novel regularizer—**D**Uality-indu**R**e*g***A**er (**DURA**)—for semantic matching knowledge graph embeddings. We first introduce the motivations in Section 4.1 and then introduce a basic version of DURA in Section 4.2. Finally, we introduce the formulation of DURA for static KGE in Section 4.3.

4.1 Motivations

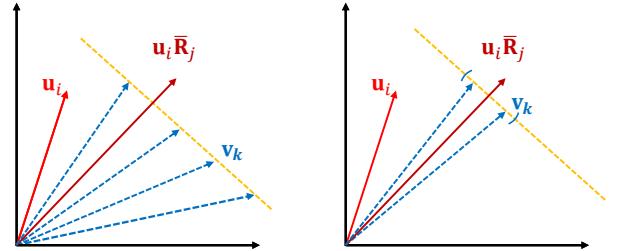
To perform accurate link prediction, we expect tail entities connected to a head entity through the same relation to have similar embeddings. First, we claim that tail entities connected to a head entity through the same relation should have similar semantics. Suppose that we know a head entity u_i and a relation r_j , and our aim is to predict the tail entity. If r_j is a one-to-many relation, i.e., there exist two entities v_1 and v_2 such that both (u_i, r_j, v_1) and (u_i, r_j, v_2) are valid, then we expect that v_1 and v_2 have similar semantics. For example, if two triples $(\text{Planck}, \text{isDoctoralAdvisorOf}, \text{Schottky})$,



(a) A KGC problem.



(b) Without regularization.



(c) With DURA.

Fig. 1: An illustration of why basic DURA can improve the performance of SM models when the embedding dimensions are 2. Suppose that triples (u_i, r_j, v_k) ($k = 1, 2, \dots, n$) are valid. (a) Figure 1a demonstrates that tail entities connected to a head entity through the same relation should have similar embeddings. (b) Figure 1b shows that SM models without regularization can get the same score even though the embeddings of v_k are dissimilar. (c) Figure 1c shows that with DURA, embeddings of v_k are encouraged to locate in a small region.

Laue) and $(\text{Planck}, \text{isDoctoralAdvisorOf}, \text{Schottky})$ are valid, then Laue and Schottky should have similar semantics. Further, we expect that entities with similar semantics have similar embeddings. In this way, if we have known that $(\text{Laue}, \text{StudiedIn}, \text{HU Berlin})$ is valid, then we can predict that $(\text{Schottky}, \text{StudiedIn}, \text{HU Berlin})$ is likely to be valid, which is consistent with the fact. See Figure 1a for an illustration of the prediction process.

However, SM models fail to achieve the above goal. As shown in Figure 1b, suppose that we have known $\mathbf{u}_i \bar{\mathbf{R}}_j$ when the embedding dimension is 2. Then, we can get the same score $s(u_i, r_j, v_k)$ for $k = 1, 2, \dots, n$, so long as \mathbf{v}_k lies on the same line perpendicular to $\mathbf{u}_i \bar{\mathbf{R}}_j$. Generally, the entities t_1 and t_2 have similar semantics. However, their embeddings \mathbf{v}_1 and \mathbf{v}_2 can even be orthogonal, which means that the two embeddings are dissimilar. Therefore, the performance of SM models for knowledge graph completion is usually unsatisfying.

4.2 Basic DURA

Consider the static knowledge graph completion problem $(u_i, r_j, ?)$. That is, we are given the head entity and the relation, aiming to predict the tail entity. Suppose that $f_j(i, k)$ measures the plausibility of a given triplet (u_i, r_j, v_k) , i.e., $f_j(i, k) = s(u_i, r_j, v_k)$. Then the score function of a SM model is

$$f_j(i, k) = \text{Re}(\bar{\mathbf{u}}_i \bar{\mathbf{R}}_j \mathbf{v}_k^\top) = \text{Re}((\mathbf{u}_i \bar{\mathbf{R}}_j, \mathbf{v}_k)). \quad (3)$$

It first maps the entity embeddings \mathbf{u}_i by a linear transformation $\bar{\mathbf{R}}_j$, and then uses the real part of an inner product to measure the similarity between $\mathbf{u}_i\bar{\mathbf{R}}_j$ and \mathbf{v}_k . Notice that another commonly used similarity measure—Euclidean distance—can replace the dot product similarity in Equation (3). We can obtain an associated distance based model formulated as

$$f_j^E(i, k) = -\|\mathbf{u}_i\bar{\mathbf{R}}_j - \mathbf{v}_k\|_2^2.$$

Note that we use the squared score function that is equivalent to the form without square. That is to say, there exists a *duality*—for an existing semantic matching KGE model (*primal*), there is often another distance based KGE model (*dual*) closely associated with it.

Specifically, the relationship between the primal and the dual can be formulated as

$$\begin{aligned} f_j^E(i, k) &= -\|\mathbf{u}_i\bar{\mathbf{R}}_j - \mathbf{v}_k\|_2^2 \\ &= -\|\mathbf{u}_i\bar{\mathbf{R}}_j\|_2^2 - \|\mathbf{v}_k\|_2^2 + 2\operatorname{Re}(\langle \mathbf{u}_i\bar{\mathbf{R}}_j, \mathbf{v}_k \rangle) \\ &= 2f_j(i, k) - \|\mathbf{u}_i\bar{\mathbf{R}}_j\|_2^2 - \|\mathbf{v}_k\|_2^2. \end{aligned}$$

When we train a distance based model, the aim is to maximize $f_j^E(i, k)$ for all valid triples (u_i, r_j, v_k) . Suppose that \mathcal{S} is the set that contains all valid triples. Then, we have that

$$\begin{aligned} \max_{(u_i, r_j, v_k) \in \mathcal{S}} f_j^E(i, k) &= \min_{(u_i, r_j, v_k) \in \mathcal{S}} -f_j^E(i, k) \\ &= \min_{(u_i, r_j, v_k) \in \mathcal{S}} -2f_j(i, k) + \|\mathbf{u}_i\bar{\mathbf{R}}_j\|_2^2 + \|\mathbf{v}_k\|_2^2. \end{aligned} \quad (4)$$

Therefore, the duality induces a regularizer for semantic matching KGE models, i.e.,

$$\sum_{(u_i, r_j, v_k) \in \mathcal{S}} \|\mathbf{u}_i\bar{\mathbf{R}}_j\|_2^2 + \|\mathbf{v}_k\|_2^2, \quad (5)$$

which is called basic DURA.

By Equation (4), we know that basic DURA constrains the distance between $\mathbf{u}_i\bar{\mathbf{R}}_j$ and \mathbf{v}_k . If \mathbf{u}_i and $\bar{\mathbf{R}}_j$ are known, then \mathbf{v}_k will lie in a small region (see Figure 1c). Thus, tail entities connected to a head entity through the same relation will have similar embeddings, which meets our expectation and is beneficial to the prediction of unknown triples.

4.3 DURA

Basic DURA encourages tail entities with similar semantics to have similar embeddings. However, it cannot handle the case that head entities have similar semantics.

Suppose that (Laue, StudiedIn, HU Berlin) and (Schottky, StudiedIn, HU Berlin) are valid. Similar to the discussion in Section 4.1, we expect that Laue and Schottky have similar semantics and thus have similar embeddings. If we further know that (DU Berlin, hasCitizen, Laue) is valid, we can predict that (DU Berlin, hasCitizen, Schottky) is valid. However, basic DURA cannot handle the case. Let $\mathbf{u}_1, \mathbf{u}_2, \mathbf{v}_1$, and \mathbf{R}_1 be the embeddings of Laue, Schottky, HU Berlin, and StudiedIn, respectively. Then, $\operatorname{Re}(\bar{\mathbf{u}}_1\mathbf{R}_1\mathbf{v}_1^\top)$ and $\operatorname{Re}(\bar{\mathbf{u}}_2\mathbf{R}_1\mathbf{v}_1^\top)$ can be equal even if \mathbf{u}_1 and \mathbf{u}_2 are orthogonal, as long as $\mathbf{u}_1\bar{\mathbf{R}}_1 = \mathbf{u}_2\bar{\mathbf{R}}_1$.

To tackle the above issue, noticing that $\operatorname{Re}(\bar{\mathbf{u}}_i\mathbf{R}_j\mathbf{v}_k^\top) = \operatorname{Re}(\bar{\mathbf{v}}_k\bar{\mathbf{R}}_j^\top\mathbf{u}_i)$, we define another dual distance based KGE model

$$\tilde{f}_j^E(i, k) = -\|\mathbf{v}_k\mathbf{R}_j^\top - \mathbf{u}_i\|_2^2.$$

Then, similar to the derivation in Equation (4), the duality induces a regularizer given by

$$\sum_{(u_i, r_j, v_k) \in \mathcal{S}} \|\mathbf{v}_k\mathbf{R}_j^\top\|_2^2 + \|\mathbf{u}_i\|_2^2. \quad (6)$$

When a SM model are incorporated with regularizer (6), head entities with similar semantics will have similar embeddings.

Finally, combining the regularizers (6) and (5), DURA has the form of

$$\sum_{(u_i, r_j, v_k) \in \mathcal{S}} \left[\|\mathbf{u}_i\bar{\mathbf{R}}_j\|_2^2 + \|\mathbf{v}_k\|_2^2 + \|\mathbf{v}_k\mathbf{R}_j^\top\|_2^2 + \|\mathbf{u}_i\|_2^2 \right]. \quad (7)$$

4.4 Theoretical Analysis for Diagonal Relation Matrices

Ignoring the sampling frequency and summarizing DURA on all possible entities and relations, we can write the regularizer as:

$$|\mathcal{E}| \sum_{j=1}^{|\mathcal{R}|} (\|\mathbf{U}\bar{\mathbf{R}}_j\|_F^2 + \|\mathbf{V}\|_F^2 + \|\mathbf{VR}_j^\top\|_F^2 + \|\mathbf{U}\|_F^2), \quad (8)$$

where $|\mathcal{E}|$ and $|\mathcal{R}|$ are the number of entities and relations, respectively.

In the rest of this section, we use the same definitions of $\hat{\mathcal{X}}_j$ and $\hat{\mathcal{X}}$ as in the problem (2). When the relation embedding matrices \mathbf{R}_j are diagonal in \mathbb{R} or \mathbb{C} as in CP or ComplEx, DURA gives an upper bound to the tensor nuclear 2-norm of $\hat{\mathcal{X}}$, which is an extension of trace norm regularizers in matrix completion. To simplify the notations, we take CP as an example, in which all involved embeddings are real numbers. The conclusion in complex space can be analogized accordingly.

Definition 1 ([10]). *The nuclear 2-norm of a 3D tensor $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \otimes \mathbb{R}^{n_3}$ is*

$$\begin{aligned} \|\mathcal{A}\|_* &= \min \left\{ \sum_{i=1}^r \|\mathbf{u}_{1,i}\|_2 \|\mathbf{u}_{2,i}\|_2 \|\mathbf{u}_{3,i}\|_2 : \right. \\ &\quad \left. \mathcal{A} = \sum_{i=1}^r \mathbf{u}_{1,i} \otimes \mathbf{u}_{2,i} \otimes \mathbf{u}_{3,i}, r \in \mathbb{N} \right\}, \end{aligned}$$

where $\mathbf{u}_{k,i} \in \mathbb{R}^{n_k}$ for $k = 1, \dots, 3$, $i = 1, \dots, r$, and \otimes denotes the outer product.

For notation convenience, we define a relation matrix $\tilde{\mathbf{R}} \in \mathbb{R}^{|\mathcal{R}| \times D}$, of which the j -th row consists of the diagonal entries of \mathbf{R}_j . That is, $\tilde{\mathbf{R}}(j, d) = \mathbf{R}_j(d, d)$, where $\mathbf{A}(i, j)$ represents the entry in the i -th row and j -th column of the matrix \mathbf{A} .

In the knowledge graph completion problem, the tensor nuclear 2-norm of $\hat{\mathcal{X}}$ is

$$\begin{aligned} \|\hat{\mathcal{X}}\|_* &= \min \left\{ \sum_{d=1}^D \|\mathbf{u}_{:,d}\|_2 \|\mathbf{r}_{:,d}\|_2 \|\mathbf{v}_{:,d}\|_2 : \right. \\ &\quad \left. \hat{\mathcal{X}} = \sum_{d=1}^D \mathbf{u}_{:,d} \otimes \mathbf{r}_{:,d} \otimes \mathbf{v}_{:,d} \right\}, \end{aligned}$$

where D is the embedding dimension, $\mathbf{u}_{:,d}$, $\mathbf{r}_{:,d}$, and $\mathbf{v}_{:,d}$ are the d -th columns of \mathbf{U} , $\tilde{\mathbf{R}}$, and \mathbf{V} .

For DURA, we have the following theorem, of which the proof are provided in Appendix A.

Theorem 1. Suppose that $\hat{\mathcal{X}}_j = \mathbf{U}\mathbf{R}_j\mathbf{V}^\top$ for $j = 1, 2, \dots, |\mathcal{R}|$, where $\mathbf{U}, \mathbf{V}, \mathbf{R}_j$ are real matrices and \mathbf{R}_j is diagonal. Then, the following equation holds

$$\begin{aligned} \min_{\hat{\mathcal{X}}_j = \mathbf{U}\mathbf{R}_j\mathbf{V}^\top} \frac{1}{4|\sqrt{\mathcal{R}}|} \sum_{j=1}^{|\mathcal{R}|} (\|\mathbf{U}\mathbf{R}_j\|_F^2 + \|\mathbf{V}\|_F^2 + \|\mathbf{V}\mathbf{R}_j^\top\|_F^2 + \|\mathbf{U}\|_F^2) \\ = \|\hat{\mathcal{X}}\|_* \end{aligned}$$

The minimizers satisfy $\|\mathbf{u}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2 = \sqrt{|\mathcal{R}|} \|\mathbf{v}_{:d}\|_2$ and $\|\mathbf{v}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2 = \sqrt{|\mathcal{R}|} \|\mathbf{u}_{:d}\|_2$, $\forall d \in \{1, 2, \dots, D\}$, where $\mathbf{u}_{:d}$, $\mathbf{r}_{:d}$, and $\mathbf{v}_{:d}$ are the d -th columns of \mathbf{U} , \mathbf{R} , and \mathbf{V} , respectively.

Therefore, the minimization of DURA has a form of the tensor nuclear 2-norm, which is a tensor analog to the matrix trace norm. Matrix trace norm regularization has shown great power in the matrix completion problem. In the experiments, we will show that DURA reproduces the success in the tensor completion problem.

5 DURA FOR TEMPORAL KGE

We first introduce a general TKGE model TRESCAL—which is an extension of RESCAL to temporal KGs—and introduce the corresponding DURA regularizer. Then, we introduce the enhanced version of DURA for TComplEx [29].

5.1 TRESCAL and DURA for TRESCAL

Inspired by the classical static KGE model RESCAL, we propose its temporal extension TRESCAL as follows:

$$s(u_i, r_j, v_k, t_l) = \bar{\mathbf{u}}_i(\mathbf{R}_j \odot \mathbf{T}_l)\mathbf{v}_k^\top, \quad (9)$$

where \mathbf{R}_j and \mathbf{T}_l are trainable real matrices. Note that the element-wise multiplication \odot is commutative, i.e., $A \odot B = B \odot A$ for all matrices A and B .

In Equation (9), the product of relation and time matrices $\mathbf{R}_j \odot \mathbf{T}_l$ can be seen as the representation of time-dependent relations, leading to $|\mathcal{R}| \times |\mathcal{T}|$ equivalent relations in a KG. By a similar derivation in Section 4.3, we propose DURA regularizer (DURA1) for TRESCAL as follows:

$$\sum_{(u_i, r_j, v_k, t_l) \in \mathcal{S}} [\|\bar{\mathbf{u}}_i(\mathbf{R}_j \odot \mathbf{T}_l)\|_2^2 + \|\mathbf{v}_k\|_2^2] \quad (10)$$

$$+ \|\mathbf{v}_k(\mathbf{R}_j \odot \mathbf{T}_l)^\top\|_2^2 + \|\mathbf{u}_i\|_2^2. \quad (11)$$

5.2 DURA for TComplEx

By introducing representations of time-dependent relations, we can extend ComplEx [23] to temporal KGs. The resulting model, TComplEx [29], is formulated as

$$s(u_i, r_j, v_k, t_l) = \mathbf{Re}(\bar{\mathbf{u}}_i(\mathbf{R}_j \odot \mathbf{T}_l)\mathbf{v}_k^\top). \quad (12)$$

As mentioned by Lacroix et al. [29], timestamp embeddings in temporal semantic matching models can be used to equivalently modulate both the relations and entities to obtain time-dependent representations. Therefore, we introduce DURA for temporal KGE in two views.

First, as relation embeddings and timestamp embeddings are diagonal matrices, the score function becomes $s(u_i, r_j, v_k, t_l) = \mathbf{Re}((\bar{\mathbf{u}}_i \mathbf{T}_l \mathbf{R}_j) \mathbf{v}_k^\top) = \mathbf{Re}(\bar{\mathbf{u}}_i(\mathbf{T}_l \mathbf{R}_j \mathbf{v}_k^\top))$. This

formulation models time-dependent relation representations. We introduce a form of DURA (DURA1) as

$$\begin{aligned} \sum_{(u_i, r_j, v_k, t_l) \in \mathcal{S}} & [\|\bar{\mathbf{u}}_i(\mathbf{R}_j \mathbf{T}_l)\|_2^2 + \|\mathbf{v}_k\|_2^2 \\ & + \|\mathbf{v}_k(\mathbf{R}_j \mathbf{T}_l)^\top\|_2^2 + \|\mathbf{u}_i\|_2^2]. \end{aligned} \quad (13)$$

Note that when the relation and time matrices are diagonal, the right hand side of Equation (12) satisfy associative law. Thus, we can reformulate the score function into $s(u_i, r_j, v_k, t_l) = \mathbf{Re}((\bar{\mathbf{u}}_i \mathbf{T}_l) \mathbf{R}_j \mathbf{v}_k^\top) = \mathbf{Re}(\bar{\mathbf{u}}_i \mathbf{R}_j(\mathbf{T}_l \mathbf{v}_k^\top))$, where $\mathbf{u}_i \mathbf{T}_l$ and $\mathbf{v}_k \mathbf{T}_l$ are representations of time-dependent entity representations. Therefore, we derive another formulation of DURA (DURA2):

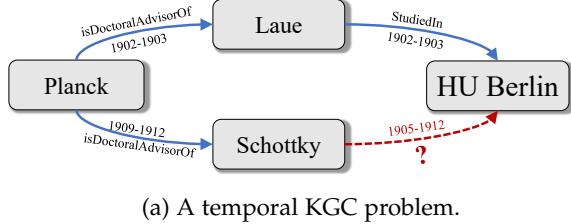
$$\begin{aligned} \sum_{(u_i, r_j, v_k, t_l) \in \mathcal{S}} & [\|\bar{\mathbf{u}}_i \mathbf{R}_j\|_2^2 + \|\mathbf{v}_k \mathbf{T}_l\|_2^2 \\ & + \|\bar{\mathbf{u}}_i \mathbf{T}_l\|_2^2 + \|\mathbf{v}_k \mathbf{R}_j\|_2^2]. \end{aligned} \quad (14)$$

We empirically evaluate the aforementioned two versions of DURA in Section 6.

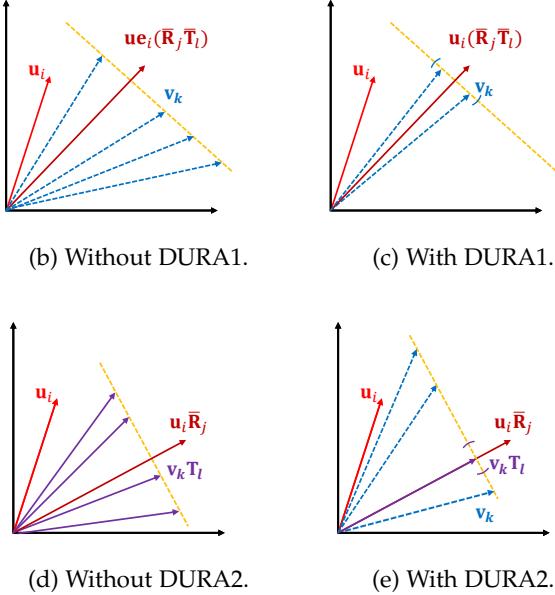
5.3 Semantic Meaning of Temporal DURA

The semantic meaning of DURA1 is similar to that of DURA for static KGE. That is, we expect tail/head entities connected to a head/tail entity through similar time-dependent relation representations to have similar embeddings. For example, as shown in Figure 2a if two quadruples (*Planck*, *isDoctoralAdvisorOf*, *Laue*, 1902–1903) and (*Planck*, *isDoctoralAdvisorOf*, *Schottky*, 1909–1912) are valid, then we expect that *Laue* and *Schottky* have similar semantics. Let $\mathbf{u}_i, \mathbf{R}_j, \mathbf{v}_k^{(1)}, \mathbf{v}_k^{(2)}, \mathbf{T}_l^{(1)}$, and $\mathbf{T}_l^{(2)}$ be embeddings of *Planck*, *isDoctoralAdvisorOf*, *Laue*, *Schottky*, 1902–1903, and 1909–1912, respectively. In temporal KGE, close timestamps usually have similar representations [29] by incorporating a timestamp smoothness regularizer. For analyses convenience and without loss of generality, we assume that $\mathbf{T}_l^{(1)} = \mathbf{T}_l^{(2)} = \mathbf{T}_l$ and hence $\bar{\mathbf{u}}_i(\mathbf{R}_j \mathbf{T}_l^{(1)}) = \bar{\mathbf{u}}_i(\mathbf{R}_j \mathbf{T}_l^{(2)}) = \bar{\mathbf{u}}_i(\mathbf{R}_j \mathbf{T}_l)$. As shown in Figure 2b, we can get the same score $s(u_i, r_j, v_k, t_l)$ for any v_k , so long as \mathbf{v}_k lies on the same line perpendicular to $\bar{\mathbf{u}}_i(\mathbf{R}_j \mathbf{T}_l)$. When we train SM models without DURA1, the embeddings $\mathbf{v}_k^{(1)}$ and $\mathbf{v}_k^{(2)}$ can even be orthogonal, which means that the two embeddings are dissimilar. In contrast, Figure 2c shows that DURA1 encourages $\mathbf{v}_k^{(1)}$ and $\mathbf{v}_k^{(2)}$ to locate in a small region, which means that two embeddings of *Laue* and *Schottky* are similar. Thus, when (*Laue*, *StudiedIn*, *HU Berlin*, 1902–1903) is valid, DURA1 encourages (*Schottky*, *StudiedIn*, *HU Berlin*, 1905–1912) to have a high score. The prediction is reasonable, as two persons advised by the same person in a similar period of time tend to study in the same university.

DURA2 encourages time-dependent entities with similar semantics to have similar embeddings, while similar embeddings of time-dependent entities do not imply similar embeddings of original entities. Similar to the above discussion, DURA2 encourages the time-dependent entities $\mathbf{v}_k^{(1)} \mathbf{T}_l$ and $\mathbf{v}_k^{(2)} \mathbf{T}_l$ to locate in a small region. However, as shown in Figure 2e, when embeddings of time-dependent entities $\mathbf{v}_k \mathbf{T}_l$ are projections of \mathbf{v}_k onto the subspace spanned



(a) A temporal KGC problem.



(d) Without DURA2.

(e) With DURA2.

Fig. 2: An illustration of how temporal DURA influence the performance of SM models when the embedding dimensions are 2. Suppose that triples (u_i, r_j, v_k, t_l) ($k = 1, 2, \dots, n$) are valid. Figure 2b and 2d show that SM models without regularization can get the same score even though the embeddings of original entities v_k and time-dependent entities $v_k T_l$ are dissimilar, respectively. Figure 2c shows that with DURA1, embeddings of v_k are encouraged to locate in a small region. Figure 2e shows that with DURA2, embeddings of time-dependent entities $v_k T_l$ are encouraged to locate in a small region, while the corresponding embeddings of original entities may be dissimilar.

by $\bar{\mathbf{u}}_i \mathbf{R}_j$, the embeddings of entities $\mathbf{v}_k^{(1)}$ and $\mathbf{v}_k^{(2)}$ can be dissimilar, even if the corresponding time-dependent entities $\mathbf{v}_k \mathbf{T}_l$ are the same. Thus, when (Laue, StudiedIn, HU Berlin, 1905-1912) is valid, (Schottky, StudiedIn, HU Berlin, 1902-1903) is not guaranteed to have a high score. However, as shown in Section 5.4, both DURA1 and DURA2 correspond to nuclear-2 norm of 4-order tensors. Thus, both of them can improve the performance of semantic matching models, while we expect DURA1 outperforms DURA2. Experiments in Section 6.2.2 confirm our points.

5.4 Theoretical Analysis for Diagonal Relation and Time Matrices

We give an analysis for the case that relation and time matrices are diagonal (i.e., TComplEx).

First, we review the definition of 4D tensors.

Definition 2 ([10]). The nuclear 2-norm of a 4D tensor $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \otimes \mathbb{R}^{n_3} \otimes \mathbb{R}^{n_4}$ is

$$\|\mathcal{A}\|_* = \min \left\{ \sum_{i=1}^r \|\boldsymbol{u}_{1,i}\|_2 \|\boldsymbol{u}_{2,i}\|_2 \|\boldsymbol{u}_{3,i}\|_2 \|\boldsymbol{u}_{4,i}\|_2 : \right.$$

$$\left. \mathcal{A} = \sum_{i=1}^r \boldsymbol{u}_{1,i} \otimes \boldsymbol{u}_{2,i} \otimes \boldsymbol{u}_{3,i} \otimes \boldsymbol{u}_{4,i}, r \in \mathbb{N} \right\},$$

where $u_{k,i} \in \mathbb{R}^{n_k}$ for $k = 1, \dots, 4$, $i = 1, \dots, r$, and \otimes denotes the outer product.

In the temporal knowledge graph completion problem, the tensor nuclear 2-norm of $\hat{\mathcal{X}}$ is

$$\|\hat{\mathcal{X}}\|_* = \min \left\{ \sum_{d=1}^D \|\mathbf{u}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2 \|\mathbf{v}_{:d}\|_2 \|\mathbf{t}_{:d}\|_2 : \right.$$

$$\left. \hat{\mathcal{X}} = \sum_{d=1}^D \mathbf{u}_{:d} \otimes \mathbf{r}_{:d} \otimes \mathbf{v}_{:d} \otimes \mathbf{t}_{:d} \right\},$$

where $\mathbf{u}_{:d}$, $\mathbf{r}_{:d}$, $\mathbf{v}_{:d}$, and $\mathbf{t}_{:d}$ are the d -th columns of the head entities matrix \mathbf{U} , the relation matrix $\tilde{\mathbf{R}}$, the tail entities matrix \mathbf{V} , and the time matrix \mathbf{V} , respectively.

Then, we have the following theorems for DURA, which correspond to the regularizers (13) and (14), respectively. The proofs are provided in the Appendixes B and C.

Theorem 2. Suppose that $\hat{X}_{jl} = U(R_j T_l) V^\top$ for $j = 1, 2, \dots, |\mathcal{R}|$, where U, V, R_j, T_l are real matrices and R_j and T_l are diagonal. Then, the following equation holds

$$\|\hat{\mathcal{X}}\|_* = \min_{\hat{\mathcal{X}}_{jl} = \mathbf{U}(\mathbf{R}_j \mathbf{T}_l) \mathbf{V}^\top} \frac{1}{4\sqrt{|\mathcal{R}||\mathcal{T}|}} \sum_{l=1}^{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{R}|} (\|\mathbf{U}\mathbf{R}_j \mathbf{T}_l\|_F^2 + \|\mathbf{V}\|_F^2 \\ + \|\mathbf{V}(\mathbf{R}_j \mathbf{T}_l)^\top\|_F^2 + \|\mathbf{U}\|_F^2).$$

The minimizers of the left side satisfy $\|\mathbf{u}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2 \|\mathbf{t}_{:d}\|_2 = \sqrt{|\mathcal{R}||\mathcal{T}|} \|\mathbf{v}_{:d}\|_2$ and $\|\mathbf{v}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2 \|\mathbf{t}_{:d}\|_2 = \sqrt{|\mathcal{R}||\mathcal{T}|} \|\mathbf{u}_{:d}\|_2$, $\forall d \in \{1, 2, \dots, D\}$, where $\mathbf{u}_{:d}$, $\mathbf{r}_{:d}$, $\mathbf{v}_{:d}$, and $\mathbf{t}_{:d}$ are the d -th columns of \mathbf{U} , \mathbf{R} , \mathbf{V} and \mathbf{V} , respectively.

Theorem 3. Suppose that $\hat{X}_{jl} = \mathbf{U}(\mathbf{R}_j \mathbf{T}_l) \mathbf{V}^\top$ for $j = 1, 2, \dots, |\mathcal{R}|$, where $\mathbf{U}, \mathbf{V}, \mathbf{R}_j, \mathbf{T}_l$ are real matrices and \mathbf{R}_j and \mathbf{T}_l are diagonal. Then, the following equation holds

$$\begin{aligned} \|\hat{\mathcal{X}}\|_* = \min_{\hat{\mathcal{X}}_{jl} = \mathbf{U}(\mathbf{R}_j \mathbf{T}_l) \mathbf{V}^\top} & \frac{1}{4\sqrt{|\mathcal{R}||\mathcal{T}|}} \sum_{l=1}^{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{R}|} (\|\mathbf{U}\mathbf{R}_j\|_F^2 + \|\mathbf{V}\mathbf{T}_l^\top\|_F^2 \\ & + \|\mathbf{U}\mathbf{T}_l\|_F^2 + \|\mathbf{V}\mathbf{R}_j^\top\|_F^2). \end{aligned}$$

The minimizers satisfy $\sqrt{|\mathcal{T}|} \|\mathbf{u}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2 = \sqrt{|\mathcal{R}|} \|\mathbf{v}_{:d}\|_2 \|\mathbf{t}_{:d}\|_2$ and $\sqrt{|\mathcal{R}|} \|\mathbf{u}_{:d}\|_2 \|\mathbf{t}_{:d}\|_2 = \sqrt{\mathcal{T}} \|\mathbf{v}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2$, $\forall d \in \{1, 2, \dots, D\}$, where $\mathbf{u}_{:d}$, $\mathbf{r}_{:d}$, and $\mathbf{t}_{:d}$ are the d -th columns of \mathbf{U} , \mathbf{R} , and \mathbf{V} , respectively.

Therefore, the minimization of DURA in the temporal knowledge graph case has a form of the tensor nuclear 2-norm of 4-order tensors. Theorem 3 also explains why DURA2 improves the performance of temporal semantic matching KGEs, even if it cannot encourage entity with similar semantics to have similar representations.

TABLE 1: Hyper-parameters found by grid search. k is the embedding size, b is the batch size, λ is the regularization coefficients, and λ_1 and λ_2 are weights for different parts of the regularizer.

	WN18RR					FB15k-237					YAGO3-10				
	k	b	λ	λ_1	λ_2	k	b	λ	λ_1	λ_2	k	b	λ	λ_1	λ_2
CP	2000	100	1e-1	0.5	1.5	2000	100	5e-2	0.5	1.5	1000	1000	5e-3	0.5	1.5
ComplEx	2000	100	1e-1	0.5	1.5	2000	100	5e-2	0.5	1.5	1000	1000	5e-2	0.5	1.5
RESCAL	512	1024	1e-1	1.0	1.0	512	512	1e-1	2.0	1.5	512	1024	5e-2	1.0	1.0

6 EXPERIMENTS

The knowledge graph completion task is a standard benchmark task to evaluate KGE models. To demonstrate that DURA is an effective and widely-applicable regularizer, we conduct extensive experiments on both static and temporal knowledge graph completion datasets.

6.1 Static Knowledge Graph Completion

We introduce the experimental settings for static KGC in Section 6.1.1 and demonstrate the effectiveness of DURA on three benchmark datasets in Section 6.1.2. Then, we compare DURA with other regularizers in Section 6.1.3.

6.1.1 Experimental Settings for Static KGC

We consider three public static knowledge graph datasets—WN18RR [40], FB15k-237 [18], and YAGO3-10 [7] for the knowledge graph completion task, which have been divided into training, validation, and testing set in previous works. The statistics of these datasets are shown in Appendix D.

We use the cross entropy loss function and the “reciprocal” setting that creates a new triplet (v_k, r_j^{-1}, u_i) for each triplet (u_i, r_j, v_k) [37], [41]. We use Adagrad [42] as the optimizer, and use grid search to find the best hyperparameters based on the performance on the validation datasets. As we regard the link prediction as a multi-class classification problem and use the cross entropy loss, we can assign different weights for different classes (i.e., tail entities) based on their frequency of occurrence in the training dataset. Specifically, suppose that the loss of a given query $(u, r, ?)$ is $\ell((u, r, ?), v)$, where v is the true tail entity, then the weighted loss is $w(t)\ell((u, r, ?), v)$, where

$$w(v) = w_0 \frac{\#v}{\max\{\#v_i : v_i \in \text{training set}\}} + (1 - w_0),$$

w_0 is a fixed number, $\#v$ denotes the frequency of occurrence in the training set of the entity v . For all models on WN18RR and RESCAL on YAGO3-10, we choose $w_0 = 0.1$ and for all the other cases we choose $w_0 = 0$.

Following [12], we use entity ranking as the evaluation task. For each triplet (u_i, r_j, v_k) in the test dataset, the model is asked to answer $(u_i, r_j, ?)$ and $(v_k, r_j^{-1}, ?)$. To do this, we fill the positions of missing entities with candidate entities to create a set of candidate triplets, and then rank the triplets in descending order by their scores. Following the “Filtered” setting in [12], we then filter out all existing triplets known to be true at ranking. We choose Mean Reciprocal Rank (MRR) and Hits at N (H@N) as the evaluation metrics. Higher MRR or H@N indicates better performance.

We find the best hyper-parameters by grid search. Specifically, we search learning rates in $\{0.1, 0.01\}$, regularization

coefficients in $\{0, 1 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}, 5 \times 10^{-2}, 1 \times 10^{-1}, 5 \times 10^{-1}\}$. On WN18RR and FB15k-237, we search batch sizes in $\{100, 500, 1000\}$ and embedding sizes in $\{500, 1000, 2000\}$. On YAGO3-10, we search batch sizes in $\{256, 512, 1024\}$ and embedding sizes in $\{500, 1000\}$. Table 1 shows the best hyperparameters for DURA found by grid search.

6.1.2 Main Results

We demonstrate the performance of DURA on several semantic matching KGE models, including CP [24], RESCAL [21], and ComplEx [23]. Note that we reimplement CP, ComplEx, and RESCAL under the “reciprocal” setting [37], [41], and obtain better results than the reported performance in the original papers. Sun et al. [43] demonstrate that many existing learning-based models use inappropriate evaluation protocols and suffer from inflating performance. Therefore, for ConvE, ConvKB, and KB-GAT, we take their results from [43] for a fair comparison.

The baseline models include distance-based models (TransE [12], TransH [8], TransD [44], RotatE [14], MuRP [30], HAKE [15]), learning-based models (ConvE [18], ConvKB [45], KB-GAT [20], InteractE [46], and STAR [47]), and semantic matching models (CP [24], RESCAL [21], ComplEx [23]). TransE is the most representative translational distance model, while failing to deal with 1-to-N, N-to-1, and N-to-N relations. TransH introduces relation-specific hyperplanes to overcome these disadvantages. TransD shares a similar idea with TransH. It introduces relation-specific spaces rather than hyperplanes. TransD also decomposes the projection matrix into a product of two vectors for further simplification. RotatE defines each relation as a rotation from the source entity to the target entity in the complex vector space and naturally, it is able to model and infer various relation patterns. MuRP embeds multi-relational graph in the Poincaré ball model of hyperbolic space to capture multiple simultaneous hierarchies. HAKE maps entities into the polar coordinate system to model semantic hierarchies, which are common in real-world applications. ConvE introduces a multi-layer convolutional network model to learn more expressive features with less parameters. ConvKB represents triples as 3-column matrixs, and employs a convolutional neural network to capture global relationships and transitional characteristics between entities and relations. KB-GAT proposes an attention-based feature embedding that captures both entity and relation features to cover the complex and hidden information in the local neighborhood surrounding a triple. InteractE improves ConvE by increasing feature interactions. StAR follows the textual encoding paradigm and augments it with graph embedding techniques.

TABLE 2: Evaluation results on WN18RR, FB15k-237 and YAGO3-10 datasets. We reimplement CP, ComplEx, and RESCAL using the “reciprocal” setting [37], [41], which leads to better results than the reported results in the original paper. * indicates that the model use external textual information.

	WN18RR			FB15k-237			YAGO3-10		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
TransE	.226	-	.501	.294	-	.465	-	-	-
TransH	.186	-	.424	.211	-	.366	-	-	-
TransD	.185	-	.428	.286	-	.453	-	-	-
RotatE	.476	.428	.571	.338	.241	.533	.495	.402	.670
MuRP	.481	.440	.566	.335	.243	.518	-	-	-
HAKE	.497	.452	.582	.346	.250	.542	.546	.462	.694
ConvE	.444	-	.503	.324	-	.501	-	-	-
ConvKB	.249	-	.524	.243	-	.421	-	-	-
KB-GAT	.412	-	.554	.157	-	.331	-	-	-
InteractE	.463	.430	.528	.354	.263	.535	.541	.462	.687
StAR	.401	.243	.709	.296	.205	.482	-	-	-
StAR (Self-Adp)*	.551	.459	.732	.365	.266	.562	-	-	-
CP	.438	.414	.485	.333	.247	.508	.567	.494	.698
RESCAL	.455	.419	.493	.353	.264	.528	.566	.490	.701
ComplEx	.460	.428	.522	.346	.256	.525	.573	.500	.703
CP-DURA	.478	.441	.552	.367	.272	.555	.579	.506	.709
RESCAL-DURA	.498	.455	.577	.368	.276	.550	.579	.505	.712
ComplEx-DURA	.491	.449	.571	.371	.276	.560	.584	.511	.713

TABLE 3: Comparison between DURA, squared Frobenius norm (FRO), and nuclear 3-norm (N3) regularizers. Results of * are taken from [37]. CP-N3 and ComplEx-N3 are re-implemented and their performances are better than the reported results in [37]. The best performance on each model are marked in bold.

	WN18RR			FB15k-237			YAGO3-10		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
CP-FRO*	.460	-	.480	.340	-	.510	.540	-	.680
CP-N3	.470	.430	.544	.354	.261	.544	.577	.505	.705
CP-DURA	.478	.441	.552	.367	.272	.555	.579	.506	.709
ComplEx-FRO*	.470	-	.540	.350	-	.530	.570	-	.710
ComplEx-N3	.489	.443	.580	.366	.271	.558	.577	.502	.711
ComplEx-DURA	.491	.449	.571	.371	.276	.560	.584	.511	.713
RESCAL-FRO	.397	.363	.452	.323	.235	.501	.474	.392	.628
RESCAL-DURA	.498	.455	.577	.368	.276	.550	.579	.505	.712

Table 2 shows the evaluation results. Overall, DURA significantly improves the performance of all considered SM models. The results demonstrate that without regularization, semantic matching models perform worse than state-of-the-art distance-based models on WN18RR, e.g., RotatE, MuRP, and HAKE. When incorporating with DURA, these semantic matching models achieve competitive performance with distance-based models. Moreover, DURA further improve the performance gain of semantic matching models on FB15k-237 and YAGO3-10. Compared with learning-based models, semantic matching models with DURA outperforms all of them without using external textual information. Notably, semantic matching models with DURA even outperform StAR (Self-Adp) on FB15k-237, which use external textual information to enhance the expressiveness of KGE. Generally, models with more parameters and datasets with smaller sizes imply a larger risk of overfitting. Among the three datasets, WN18RR has the smallest size of only 11 kinds of relations and around 80k training samples. Therefore, the improvements brought by DURA on WN18RR are expected to be larger compared with other datasets, which is consistent with the experiments. As

stated in [48], RESCAL is a more expressive model, but it is prone to overfit on small- and medium-sized datasets because it represents relations with much more parameters. For example, on WN18RR dataset, RESCAL gets an H@10 score of 0.493, which is lower than ComplEx (0.522). The advantage of its expressiveness does not show up at all. Incorporated with DURA, RESCAL gets an 8.4% improvement on H@10 and finally attains 0.577, outperforming all compared models. On larger datasets such as YAGO3-10, overfitting also exists but will be insignificant. Nonetheless, DURA still leads to consistent improvement, showing the ability of DURA to prevent models from overfitting.

6.1.3 Comparison with Other Regularizers

We compare DURA to the squared Frobenius norm regularizer and the tensor nuclear 3-norm (N3) regularizer [37]. The squared Frobenius norm regularizer $g(\hat{\mathcal{X}}) = \|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 + \sum_{j=1}^{|\mathcal{R}|} \|\mathbf{R}_j\|_F^2$. N3 regularizer is given by $g(\hat{\mathcal{X}}) = \sum_{d=1}^D (\|\mathbf{u}_{:d}\|_3^3 + \|\mathbf{r}_{:d}\|_3^3 + \|\mathbf{v}_{:d}\|_3^3)$, where $\|\cdot\|_3$ denotes L_3 norm of vectors. N3 regularizer is only suitable for models in which \mathbf{R}_j is diagonal, such as CP or ComplEx.

TABLE 4: Hyper-parameters found by grid search. λ is the regularization coefficients, λ_1 and λ_2 are weights for different parts of the regularizer for DURA2, and λ_3 and λ_4 are weights for different parts of the regularizer for DURA1.

	ICEWS14					ICEWS05-15					YAGO15k				
	λ	λ_1	λ_2	λ_3	λ_4	λ	λ_1	λ_2	λ_3	λ_4	λ	λ_1	λ_2	λ_3	λ_4
TComplEx-DURA1	1e-2	-	-	1e-3	1e2	1e-3	-	-	3e-2	30	1e-2	-	-	1e-2	1e2
TComplEx-DURA2	1e-1	1e-1	3e-2	-	-	1e-1	1e-2	3e-2	-	-	1e-2	3e-2	3e-2	-	-
TRESCAL-DURA1	1e-2	-	-	1e1	1e1	1e-2	-	-	1e1	1	1e-2	-	-	1e-1	1e1

We implement both the squared Frobenius norm (FRO) and N3 regularizers in the weighted way as stated in [37]. Table 3 shows the performance of the three regularizers on three popular models: CP, ComplEx, and RESCAL. Note that when the considered model is RESCAL, we only compare DURA to the squared Frobenius norm regularization as N3 does not apply to it.

For CP and ComplEx, DURA brings consistent improvements compared to FRO and N3 on all datasets. Specifically, on FB15k-237, compared to CP-N3, CP-DURA gets an improvement of 0.013 in terms of MRR. Even for the previous state-of-the-art SM model ComplEx, DURA brings further improvements against the N3 regularizer. Incorporated with FRO, RESCAL performs worse than the vanilla model, which is consistent with the results in [49]. However, RESCAL-DURA brings significant improvements against RESCAL. All the results demonstrate that DURA is more widely applicable than N3 and more effective than the squared Frobenius norm regularizer.

6.2 Temporal Knowledge Graph Completion

We also conduct extensive experiments on temporal knowledge graph completion datasets. Specifically, we introduce the experimental settings for temporal KGC in Section 6.2.1 and show the effectiveness of DURA on three benchmark datasets in Section 6.2.2.

6.2.1 Experimental Settings for Temporal KGC

We consider three public temporal knowledge graph datasets—ICEWS14 [50], ICEWS05-15 [50], and YAGO15k [51] for the knowledge graph completion task, which have been divided into training, validation, and testing set in previous works. The statistics of these datasets are shown in Appendix D.

Following [29], we use the following temporal regularizer for TComplEx to smooth timestamp embeddings

$$\Lambda_3(\mathcal{T}) = \frac{1}{|\mathcal{T}| - 1} \sum_{l=1}^{|\mathcal{T}| - 1} \|\text{diag}^{-1}(\mathbf{T}_{l+1} - \mathbf{T}_l)\|_3^3,$$

where $\text{diag}^{-1}(\cdot)$ gives the vector made of the diagonal elements from the a matrix. For TRESCAL, we propose to use the temporal regularizer

$$\Lambda_2(\mathcal{T}) = \frac{1}{|\mathcal{T}| - 1} \sum_{l=1}^{|\mathcal{T}| - 1} \|\mathbf{T}_{l+1} - \mathbf{T}_l\|_F^2$$

to smooth timestamp embeddings.

Moreover, we find it better to assign different weights for the parts involved with relations. That is, the optimization problem has the form of

$$\begin{aligned} \min \sum_{(u_i, r_j, v_k, t_l) \in \mathcal{S}} & [\ell_{ijkl}(\mathbf{U}, \mathbf{R}_1, \dots, \mathbf{R}_J, \mathbf{V}, \mathbf{T}_1, \dots, \mathbf{T}_L) \\ & + \lambda(\|\bar{\mathbf{u}}_i \mathbf{R}_j\|_2^2 + \|\mathbf{v}_k \mathbf{T}_l\|_2^2) \\ & + \lambda_2(\|\bar{\mathbf{u}}_i \mathbf{T}_l\|_2^2 + \|\mathbf{v}_k \mathbf{R}_j\|_2^2) \\ & + \lambda_3(\|\mathbf{u}_i\|_2^2 + \|\mathbf{v}_k\|_2^2) \\ & + \lambda_4(\|\bar{\mathbf{u}}_i (\mathbf{R}_j \odot \mathbf{T}_l)\|_2^2 + \|\mathbf{v}_k (\mathbf{R}_j \odot \mathbf{T}_l)\|_2^2)] \end{aligned}$$

where $\lambda, \lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$ are fixed hyperparameters. We denote the regularizer (13) by DURA2 (i.e., $\lambda_3 = \lambda_4 = 0$) and the regularizer (14) by DURA1 (i.e., $\lambda_1 = \lambda_2 = 0$).

We use grid search to find the best hyperparameters based on the performance on the validation datasets. We search $\lambda \in \{1, 3e-1, 1e-1, 3e-2, 1e-2, 3e-3, 1e-3, 3e-4, 1e-4\}$ for all experiments. We search $\lambda_1, \lambda_2 \in \{1, 3e-1, 1e-1, 3e-2, 1e-2, 3e-3, 1e-3\}$ for DURA2, and $\lambda_3 \in \{1e2, 3e1, 1e1, 3, 1, 3e-1, 1e-1, 3e-2, 1e-2, 3e-3, 1e-3, 3e-4, 1e-4\}$, $\lambda_4 \in \{1e3, 3e2, 1e2, 3e1, 1e1, 3, 1, 3e-1, 1e-1, 3e-2\}$ for DURA1. Table 4 shows the best hyperparameters found by grid search.

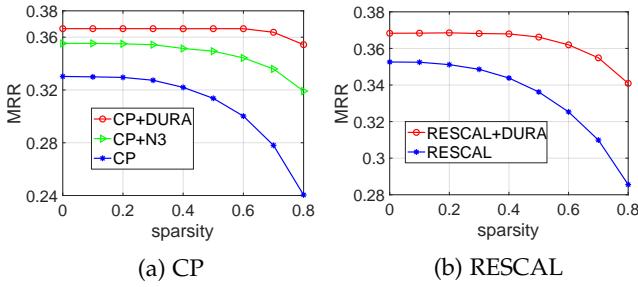
6.2.2 Main Results on Temporal KGC

We compare DURA with the popular squared Frobenius norm and the tensor nuclear 3-norm (N3) regularizers [29] on the state-of-the-art temporal semantic matching KGE model TComplEx and our proposed TRESCAL method. We also include four recent distance-based models, including TTransE [16], TA-TransE [52], RFTE-HAKE [53] and BoxTE [54]. TTransE [16] proposes a time-aware knowledge graph completion model by using temporal order information among facts. TA-TransE [52] utilizes recurrent neural networks to learn time-aware representations. RFTE [53] is a framework to transplant static KGE models to temporal KGE models. It treats the sequence of graphs as a Markov chain. BoxTE [54] assumes that time is a filter that helps pick out answers to be correct during certain periods. Therefore, it introduces boxes to represent a set of answer entities to a time-agnostic query.

Table 5 shows the effectiveness of DURA. Note that DURA2 as shown in the formulation (14) does not applicable to TRESCAL since the matrix multiplication and element-wise multiplication are not commutative. Moreover, the N3 regularizer is not applicable to TRESCAL since this model is not CP tensor factorization. Table 5 shows that DURA effectively improve the performance of TComplEx and TRESCAL. With DURA, these semantic marching models outperform the state-of-the-art distance-based models in terms of MRR.

TABLE 5: Evaluation results on ICEWS14, ICEWS05-15, and YAGO15k datasets.

	ICEWS14			ICEWS05-15			YAGO15k		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
TTransE	.26	.07	.60	.27	.08	.62	.32	.23	.51
TA-TransE	.28	.10	.63	.30	.10	.67	.32	.23	.51
RFTE-HAKE	.50	.40	.70	.47	.37	.65	-	-	-
BoxTE (k=2)	.62	.53	.77	.66	.58	.82	-	-	-
TComplEx	.60	.51	.75	.65	.57	.79	.35	.27	.52
TComplEx-FRO	.60	.53	.74	.65	.57	.80	.34	.27	.49
TComplEx-N3	.61	.53	.77	.66	.59	.80	.36	.28	.54
TComplEx-DURA1	.64	.56	.79	.67	.59	.81	.38	.30	.55
TComplEx-DURA2	.62	.54	.77	.67	.59	.81	.35	.27	.52
TRESCAL	.56	.47	.73	.65	.57	.79	.28	.21	.45
TRESCAL-FRO	.55	.48	.69	.60	.52	.75	.32	.25	.46
TRESCAL-DURA1	.60	.51	.76	.67	.60	.80	.30	.22	.48

Fig. 3: The effect of entity embeddings' λ -sparsity on MRR. The experiments are conducted on FB15k-237.

6.3 Sparsity Analysis

As knowledge graphs usually contain billions of entities, the storage of entity embeddings faces severe challenges. Intuitively, if embeddings are sparse, that is, most of the entries are zero, we can store them with less storage. Therefore, the sparsity of the generated entity embeddings becomes crucial for real-world applications. Generally, there are few entries of entity embeddings that are exactly equal to 0 after training, which means that it is hard to obtain sparse entity embeddings directly. However, when we score triples using the trained model, the embedding entries with values close to 0 will have minor contributions to the score of a triplet. If we set the embedding entries close to 0 to be exactly 0, we could transform the embeddings into sparse ones. Therefore, there is a trade-off between the sparsity and performance decrement. We define the following λ -sparsity to indicate the proportion of entries that are close to zero:

$$s_\lambda = \frac{\sum_{i=1}^I \sum_{d=1}^D \mathbb{1}_{\{|x| < \lambda\}}(\mathbf{E}_{id})}{I \times D}, \quad (15)$$

where $\mathbf{E} \in \mathbb{R}^{I \times D}$ is the entity embedding matrix, \mathbf{E}_{id} is the entry in the i -th row and d -th column of \mathbf{E} , I is the number of entities, D is the embedding dimension, and $\mathbb{1}_C(x)$ is the indicator function taking value of 1 if $x \in C$ or otherwise 0.

Figure 3 shows the effect of entity embeddings' λ -sparsity on MRR. Following Equation (15), we select entries of which the absolute value are less than a threshold and set them to be 0. Note that for any given s_λ , we can always find a proper threshold λ to approximate it, as the formula

TABLE 6: Ablation results on the combined and uncombined regularizers. “-I” and “-II” represents the regularizer defined in formula (5) and (6), respectively.

	WN18RR		FB15k-237	
	MRR	H@10	MRR	H@10
CP	.438	.485	.333	.508
CP-DURA-I	.463	.535	.331	.522
CP-DURA-II	.439	.492	.332	.507
CP-DURA	.478	.552	.367	.555
RESCAL	.455	.493	.353	.528
RESCAL-DURA-I	.475	.541	.341	.523
RESCAL-DURA-II	.462	.524	.342	.510
RESCAL-DURA	.498	.577	.368	.550

is increasing with respect to λ . Results in the figure show that DURA causes much gentler performance decrement as the embedding sparsity increases. In Figure 3a, incorporated with DURA, CP maintains MRR of 0.366 unchanged even when 60% entries are set to 0. More surprisingly, when the sparsity reaches 70%, CP-DURA can still outperform CP-N3 with zero sparsity. For RESCAL, when set 80% entries to be 0, RESCAL+DURA still has the MRR of 0.341, which significantly outperforms vanilla RESCAL, whose MRR has decreased from 0.352 to 0.286. In a word, incorporating with DURA regularizer, the performance of CP and RESCAL remains comparable to the state-of-the-art models, even when 70% of entity embeddings' entries are set to 0.

Following [55], we store the sparse embedding matrices using compressed sparse row (CSR) or compressed sparse column (CSC) format. Experiments show that DURA brings about 65% fewer storage costs for entity embeddings when 70% of the entries are set to 0.

6.4 Ablation Study

We conduct ablation studies to show the effectiveness of the combination of (5) and (6). The results are shown in Table 6, where “-I” and “-II” represent the regularizer defined in formula (5) and (6), respectively. On WN18RR, the uncombined regularizer DURA-I and DURA-II can also improve the performance for the semantic matching models, but the combination brings further improvements. On FB15k-237, DURA-I and DURA-II even lead to a decrement in terms of MRR. One possible reason is that FB15k-237

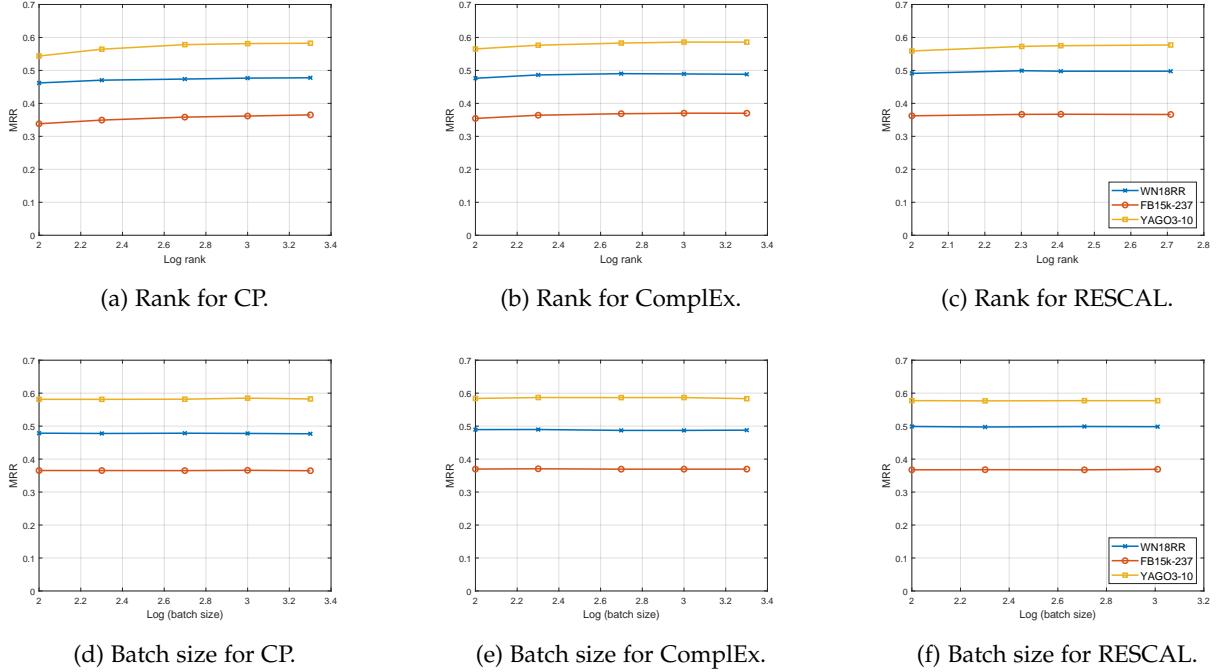


Fig. 4: Sensitivity to hyper-parameters of DURA. The x-axis is the logarithm of each hyper-parameter based on 10 and The y-axis is mean reciprocal rank (MRR) on test data.

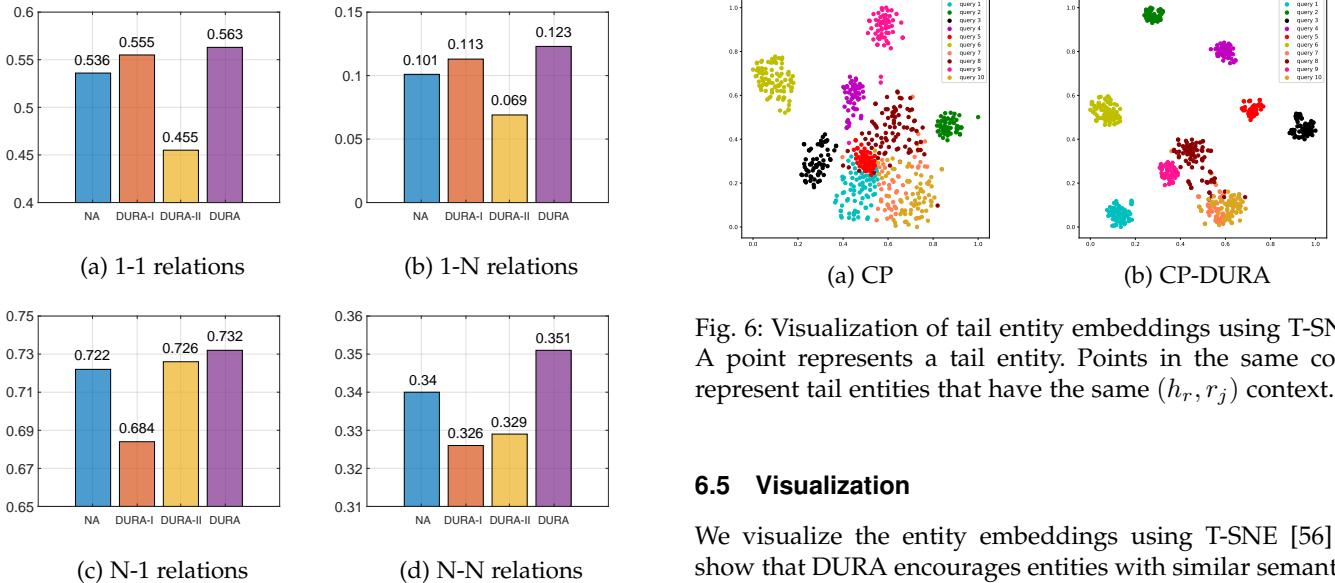


Fig. 5: Average MRR of RESCAL with different regularizers on different types of relations. The dataset is FB15k-237.

has a large number of 1-N and N-1 relations (refer to [8] for the definitions relation types) and they have different preferences for the uncombined regularizers. Figure 5 shows the performance of RESCAL with different regularizers on different relation types. Compared with vanilla RESCAL, DURA-I only improves the MRR for 1-1 and 1-N relations, while DURA-II only improves the MRR for N-1 relations, which validates our analysis. DURA significantly outperforms both DURA-I and DURA-II on all relation types.

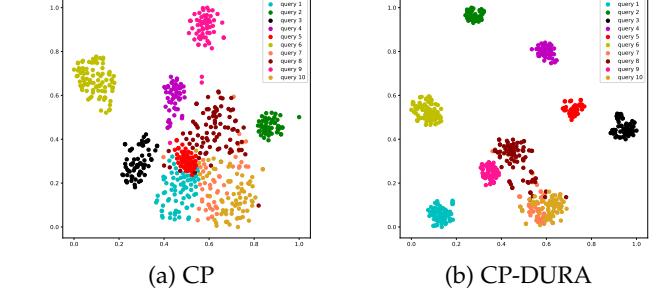


Fig. 6: Visualization of tail entity embeddings using T-SNE. A point represents a tail entity. Points in the same color represent tail entities that have the same (h_r, r_j) context.

6.5 Visualization

We visualize the entity embeddings using T-SNE [56] to show that DURA encourages entities with similar semantics to have similar embeddings. Without loss of generality, we use tail entities for visualization.

Suppose that (u_i, r_j) is a *query*, where u_i and r_j are head entities and relations, respectively. An entity v_k is an *answer* to a query (u_i, r_j) if (u_i, r_j, v_k) is valid. We randomly selected 10 queries in FB15k-237, each of which has more than 50 answers. Then, we use T-SNE to visualize the answers' embeddings generated by CP and CP-DURA. Figure 6 shows the visualization results. Each entity is represented by a 2D point and points in the same color represent tail entities with the same (u_i, r_j) context (i.e. query). Figure 6 shows that, with DURA, entities with the same (u_i, r_j) contexts are indeed being assigned more similar representations, which verifies the claims in Section 4.1.

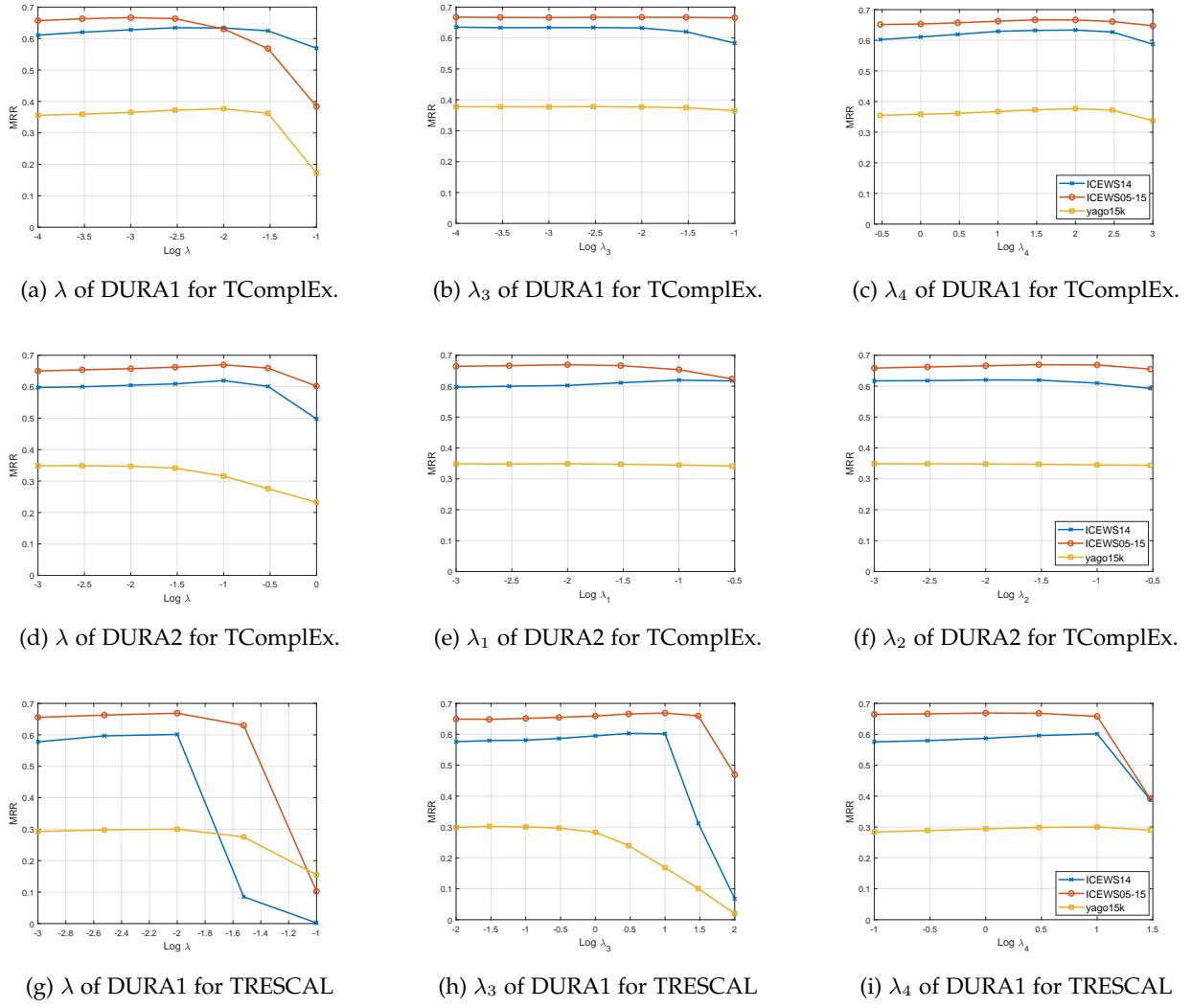


Fig. 7: Sensitivity to hyper-parameters of DURA1 and DURA2. The x-axis is the logarithm of each hyper-parameter based on 10 and the y-axis is mean reciprocal rank (MRR) on test data.

6.6 Sensitivity to Hyper-Parameters

For static KGE, the hyper-parameters mainly include embedding sizes k , batch sizes b , and regularization coefficients $\lambda, \lambda_i (i = 1, 2)$; for temporal KGE, the hyper-parameters mainly include regularization coefficients $\lambda, \lambda_i (i = 1, 2, 3, 4)$. We provide the performance curves on different datasets for these hyper-parameters. Note that when exploring the effect of a specific hyper-parameter, we fix the other hyper-parameters as their best values.

As shown in Figure 4, the performance is stable when the embedding and batch sizes vary. Figure 8 demonstrates that the performance of static KGE is relatively insensitive to λ_i . However, when the regularization coefficients λ are too large, the performance drops quickly. It is expectable since too strong regularization will weaken models' ability to model complex relations.

As shown in Figure 7, the performance of temporal KGE is relatively insensitive to λ and λ_i when the KGE model is TComplEx. However, the performance of TRESCAL drops quickly when these regularization coefficients are large.

It suggests that we should choose hyperparameters for TRESCAL carefully.

7 CONCLUSION

We propose a widely applicable and effective regularizer—namely, DURA—for semantic matching knowledge graph embedding models. The formulation of DURA is based on the observation that, for an existing semantic matching KGE model (primal), there is often another distance based KGE model (dual) closely associated with it. Extensive experiments show that DURA yields consistent and significant improvements on both static and temporal knowledge graph embedding benchmark datasets.

ACKNOWLEDGMENT

This work was supported in part by National Science Foundations of China grants 61822604, U19B2026, 61836006, and 62021001, and the Fundamental Research Funds for the Central Universities grant WK3490000004.

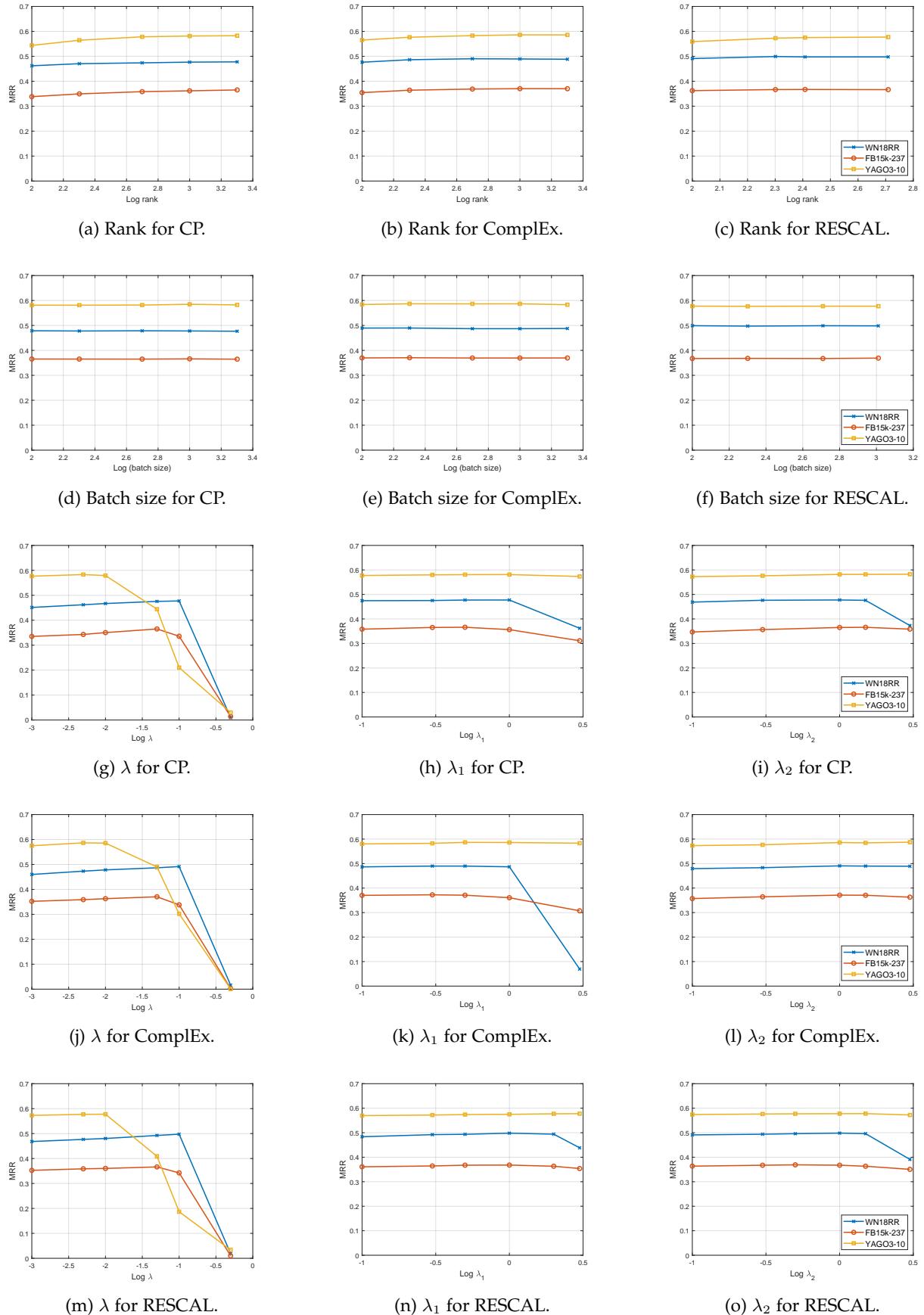


Fig. 8: Sensitivity to hyper-parameters of DURA. The x-axis is the logarithm of each hyper-parameter based on 10 and the y-axis is mean reciprocal rank (MRR) on test data.

REFERENCES

- [1] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1441–1451.
- [2] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based question answering," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, p. 105–113.
- [3] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, p. 417–426.
- [4] K. Marino, R. Salakhutdinov, and A. K. Gupta, "The more you know: Using knowledge graphs for image classification," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 20–28, 2017.
- [5] 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*, 2008, p. 1247–1250.
- [6] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.
- [7] F. Mahdisoltani, J. Biega, and F. M. Suchanek, "YAGO3: A knowledge base from multilingual wikipedias," in *Seventh Biennial Conference on Innovative Data Systems Research*, 2015.
- [8] 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.
- [9] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, p. 2181–2187.
- [10] S. Friedland and L.-H. Lim, "Nuclear norm of higher-order tensors," *Mathematics of Computation*, vol. 87, no. 311, pp. 1255–1281, 2018.
- [11] Z. Zhang, J. Cai, and J. Wang, "Duality-induced regularizer for tensor factorization based knowledge graph completion," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 21604–21615.
- [12] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems 26*, 2013, pp. 2787–2795.
- [13] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011, p. 301–306.
- [14] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," in *International Conference on Learning Representations*, 2019.
- [15] Z. Zhang, J. Cai, Y. Zhang, and J. Wang, "Learning hierarchy-aware knowledge graph embeddings for link prediction," in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.
- [16] T. Jiang, T. Liu, T. Ge, L. Sha, B. Chang, S. Li, and Z. Sui, "Towards time-aware knowledge graph completion," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Dec. 2016, pp. 1715–1724.
- [17] S. S. Dasgupta, S. N. Ray, and P. Talukdar, "HyTE: Hyperplane-based temporally aware knowledge graph embedding," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Oct.–Nov. 2018, pp. 2001–2011.
- [18] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, 2018, pp. 1811–1818.
- [19] D. Q. Nguyen, T. Vu, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization," in *NAAACL*, 2019.
- [20] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning attention-based embeddings for relation prediction in knowledge graphs," *arXiv preprint arXiv:1906.01195*, 2019.
- [21] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proceedings of the 28th International Conference on Machine Learning*, vol. 11, 2011, pp. 809–816.
- [22] B. Yang, S. W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *Proceedings of the International Conference on Learning Representations*, 2015.
- [23] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proceedings of The 33rd International Conference on Machine Learning*, vol. 48, 2016, pp. 2071–2080.
- [24] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Journal of Mathematics and Physics*, vol. 6, no. 1–4, pp. 164–189, 1927.
- [25] Y. Ma, V. Tresp, and E. A. Daxberger, "Embedding models for episodic knowledge graphs," *Journal of Web Semantics*, vol. 59, p. 100490, 2019.
- [26] 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*, 2019, pp. 5185–5194.
- [27] B. W. Bader, R. A. Harshman, and T. G. Kolda, "Temporal analysis of semantic graphs using alsasan," in *Seventh IEEE international conference on data mining*, 2007, pp. 33–42.
- [28] R. A. Harshman, P. E. Green, Y. Wind, and M. E. Lundy, "A model for the analysis of asymmetric data in marketing research," *Marketing Science*, vol. 1, no. 2, pp. 205–242, 1982.
- [29] T. Lacroix, G. Obozinski, and N. Usunier, "Tensor decompositions for temporal knowledge base completion," *arXiv preprint arXiv:2004.04926*, 2020.
- [30] I. Balazevic, C. Allen, and T. Hospedales, "Multi-relational poincaré graph embeddings," *Advances in Neural Information Processing Systems*, vol. 32, pp. 4463–4473, 2019.
- [31] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, p. 855–864.
- [32] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, p. 701–710.
- [33] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, 2015, p. 1067–1077.
- [34] M. Ye, J. Shen, X. Zhang, P. C. Yuen, and S.-F. Chang, "Augmentation invariant and instance spreading feature for softmax embedding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 924–939, 2022.
- [35] A. Van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.
- [36] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, 13–18 Jul 2020, pp. 1597–1607.
- [37] T. Lacroix, N. Usunier, and G. Obozinski, "Canonical tensor decomposition for knowledge base completion," in *Proceedings of the International Conference on Machine Learning*, 2018, pp. 2863–2872.
- [38] P. Minervini, L. Costabello, E. Muñoz, V. Nováček, and P.-Y. Vandebussche, "Regularizing knowledge graph embeddings via equivalence and inversion axioms," in *Machine Learning and Knowledge Discovery in Databases*, 2017, pp. 668–683.
- [39] B. Ding, Q. Wang, B. Wang, and L. Guo, "Improving knowledge graph embedding using simple constraints," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jul. 2018, pp. 110–121.
- [40] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *3rd Workshop on Continuous Vector Space Models and Their Compositionality*, 2015.
- [41] S. M. Kazemi and D. Poole, "Simple embedding for link prediction in knowledge graphs," in *Advances in Neural Information Processing Systems 31*, 2018, pp. 4284–4295.
- [42] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.
- [43] Z. Sun, S. Vashishth, S. Sanyal, P. Talukdar, and Y. Yang, "A re-evaluation of knowledge graph completion methods," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Jul. 2020, pp. 5516–5522.

- [44] 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.
- [45] 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 *NAACL*, 2018.
- [46] S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal, and P. Talukdar, "Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020, pp. 3009–3016.
- [47] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, and Y. Chang, "Structure-augmented text representation learning for efficient knowledge graph completion," in *Proceedings of the Web Conference*, 2021, pp. 1737–1748.
- [48] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.
- [49] D. Ruffinelli, S. Broscheit, and R. Gemulla, "You can teach an old dog new tricks! on training knowledge graph embeddings," in *International Conference on Learning Representations*, 2020.
- [50] E. Boschee, J. Lautenschlager, S. O'Brien, S. Shellman, J. Starz, and M. Ward, "ICEWS Coded Event Data," 2015.
- [51] A. García-Durán, S. Dumančić, and M. Niepert, "Learning sequence encoders for temporal knowledge graph completion," *arXiv preprint arXiv:1809.03202*, 2018.
- [52] A. García-Durán, S. Dumančić, and M. Niepert, "Learning sequence encoders for temporal knowledge graph completion," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Oct.-Nov. 2018, pp. 4816–4821.
- [53] Y. Xu, E. Haihong, M. Song, W. Song, X. Lv, W. Haotian, and Y. Jinrui, "Rtfe: a recursive temporal fact embedding framework for temporal knowledge graph completion," *arXiv preprint arXiv:2009.14653*, 2020.
- [54] J. Messner, R. Abboud, and I. I. Ceylan, "Temporal knowledge graph completion using box embeddings," *arXiv preprint arXiv:2109.08970*, 2021.
- [55] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *International Conference on Learning Representations*, 2015.
- [56] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.



Zhihao Shi received the B.Sc. degree in Department of Electronic Engineering and Information Science, Hefei, China, in 2020. He is currently a graduate student in the Department of Electronic Engineering and Information Science at University of Science and Technology of China, Hefei, China. His research interests include graph representation learning and reinforcement learning.



Jianyu Cai received the B.Sc. degree in software engineering from Southeast University, Nanjing, China, in 2019. He is currently a graduate student in the Department of Electronic Engineering and Information Science at University of Science and Technology of China, Hefei, China. His research interests include knowledge graph and natural language processing.



Shuiwang Ji received the PhD degree in computer science from Arizona State University, Tempe, Arizona, in 2010. Currently, he is an Associate Professor in the Department of Computer Science and Engineering, Texas A&M University, College Station, Texas. His research interests include machine learning, deep learning, data mining, and computational biology. He received the National Science Foundation CAREER Award in 2014. He is currently an Associate Editor for *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *ACM Transactions on Knowledge Discovery from Data*, and *ACM Computing Surveys*. He regularly serves as an Area Chair or equivalent roles for data mining and machine learning conferences, including AAAI, ICLR, ICML, IJCAI, KDD, and NeurIPS. He is a senior member of IEEE.



Jie Wang received the B.Sc. degree in electronic information science and technology from University of Science and Technology of China, Hefei, China, in 2005, and the Ph.D. degree in computational science from the Florida State University, Tallahassee, FL, in 2011. He is currently a professor in the Department of Electronic Engineering and Information Science at University of Science and Technology of China, Hefei, China. His research interests include reinforcement learning, knowledge graph, large-scale optimization, deep learning, etc. He is a senior member of IEEE.



Feng Wu received the B.S. degree in electrical engineering from Xidian University in 1992, and the M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology in 1996 and 1999, respectively. He is currently a Professor with the University of Science and Technology of China, where he is also the Dean of the School of Information Science and Technology. Before that, he was a Principal Researcher and the Research Manager with Microsoft Research Asia. His research interests include image and video compression, media communication, and media analysis and synthesis. He has authored or coauthored over 200 high quality articles (including several dozens of IEEE Transaction papers) and top conference papers on MOBICOM, SIGIR, CVPR, and ACM MM. He has 77 granted U.S. patents. His 15 techniques have been adopted into international video coding standards. As a coauthor, he received the Best Paper Award at 2009 IEEE Transactions on Circuits and Systems for Video Technology, PCM 2008, and SPIE VCIP 2007. He also received the Best Associate Editor Award from IEEE Circuits and Systems Society in 2012. He also serves as the TPC Chair for MMSP 2011, VCIP 2010, and PCM 2009, and the Special Sessions Chair for ICME 2010 and ISCAS 2013. He serves as an Associate Editor for *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions ON Multimedia*, and several other international journals.



Zhanqiu Zhang received the B.Sc. degree in information and computing science from University of Science and Technology of China, Hefei, China, in 2018. He is currently a Ph.D. candidate in the Department of Electronic Engineering and Information Science at University of Science and Technology of China, Hefei, China. His research interests include graph representation learning and natural language processing.

APPENDIX A PROOF OF THEOREM 1

Proof. We have that

$$\begin{aligned}
& \sum_{j=1}^{|\mathcal{R}|} (\|\mathbf{U}\mathbf{R}_j\|_F^2 + \|\mathbf{V}\|_F^2) \\
&= \sum_{j=1}^{|\mathcal{R}|} \left(\sum_{i=1}^I \|\mathbf{u}_i \circ \mathbf{r}_j\|_F^2 + \sum_{d=1}^D \|\mathbf{v}_{:d}\|_F^2 \right) \\
&= \sum_{j=1}^{|\mathcal{R}|} \left(\sum_{d=1}^D \|\mathbf{v}_{:d}\|_F^2 + \sum_{i=1}^I \sum_{d=1}^D \mathbf{u}_{id}^2 \mathbf{r}_{jd}^2 \right) \\
&= \sum_{j=1}^{|\mathcal{R}|} \sum_{d=1}^D \|\mathbf{v}_{:d}\|_2^2 + \sum_{d=1}^D \|\mathbf{u}_{:d}\|_2^2 \|\mathbf{r}_{:d}\|_2^2 \\
&= \sum_{d=1}^D (\|\mathbf{u}_{:d}\|_2^2 \|\mathbf{r}_{:d}\|_2^2 + |\mathcal{R}| \|\mathbf{v}_{:d}\|_2^2) \\
&\geq \sum_{d=1}^D 2\sqrt{|\mathcal{R}|} \|\mathbf{u}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2 \|\mathbf{v}_{:d}\|_2 \\
&= 2\sqrt{|\mathcal{R}|} \sum_{d=1}^D \|\mathbf{u}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2 \|\mathbf{v}_{:d}\|_2.
\end{aligned}$$

The equality holds if and only if $\|\mathbf{u}_{:d}\|_2^2 \|\mathbf{r}_{:d}\|_2^2 = |\mathcal{R}| \|\mathbf{v}_{:d}\|_2^2$, i.e., $\|\mathbf{u}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2 = \sqrt{|\mathcal{R}|} \|\mathbf{v}_{:d}\|_2$.

For all CP decomposition $\hat{\mathcal{X}} = \sum_{d=1}^D \mathbf{u}_{:d} \otimes \mathbf{r}_{:d} \otimes \mathbf{v}_{:d}$, we can always let $\mathbf{u}'_{:d} = \mathbf{u}_{:d}$, $\mathbf{r}'_{:d} = \sqrt{\frac{\|\mathbf{v}_{:d}\|_2 \sqrt{|\mathcal{R}|}}{\|\mathbf{u}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2}} \mathbf{r}_{:d}$ and $\mathbf{v}'_{:d} = \sqrt{\frac{\|\mathbf{u}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2}{\|\mathbf{v}_{:d}\|_2 \sqrt{|\mathcal{R}|}}} \mathbf{v}_{:d}$ such that

$$\|\mathbf{u}'_{:d}\|_2 \|\mathbf{r}'_{:d}\|_2 = \sqrt{|\mathcal{R}|} \|\mathbf{v}'_{:d}\|_2,$$

and meanwhile ensure that $\hat{\mathcal{X}} = \sum_{d=1}^D \mathbf{u}'_{:d} \otimes \mathbf{r}'_{:d} \otimes \mathbf{v}'_{:d}$. Therefore, we know that

$$\begin{aligned}
\frac{1}{\sqrt{|\mathcal{R}|}} \sum_{j=1}^{|\mathcal{R}|} \|\hat{\mathcal{X}}_j\|_* &= \frac{1}{2\sqrt{|\mathcal{R}|}} \sum_{j=1}^{|\mathcal{R}|} \min_{\hat{\mathcal{X}}_j = \mathbf{U}\mathbf{R}_j\mathbf{V}^\top} (\|\mathbf{U}\mathbf{R}_j\|_F^2 + \|\mathbf{V}\|_F^2) \\
&\leq \frac{1}{2\sqrt{|\mathcal{R}|}} \min_{\hat{\mathcal{X}}_j = \mathbf{U}\mathbf{R}_j\mathbf{V}^\top} \sum_{j=1}^{|\mathcal{R}|} (\|\mathbf{U}\mathbf{R}_j\|_F^2 + \|\mathbf{V}\|_F^2) \\
&= \min_{\hat{\mathcal{X}} = \sum_{d=1}^D \mathbf{u}_{:d} \otimes \mathbf{r}_{:d} \otimes \mathbf{v}_{:d}} \sum_{d=1}^D \|\mathbf{u}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2 \|\mathbf{v}_{:d}\|_2 \\
&= \|\hat{\mathcal{X}}\|_*.
\end{aligned}$$

In the same manner, we know that

$$\frac{1}{2\sqrt{|\mathcal{R}|}} \min_{\hat{\mathcal{X}}_j = \mathbf{U}\mathbf{R}_j\mathbf{V}^\top} \sum_{j=1}^{|\mathcal{R}|} (\|\mathbf{V}\mathbf{R}_j^\top\|_F^2 + \|\mathbf{U}\|_F^2) = \|\hat{\mathcal{X}}\|_*$$

The minimizers satisfy $\|\mathbf{v}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2 = \sqrt{|\mathcal{R}|} \|\mathbf{u}_{:d}\|_2$.

Therefore, the minimizers satisfy $\|\mathbf{u}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2 = \sqrt{|\mathcal{R}|} \|\mathbf{v}_{:d}\|_2$, $\forall d \in \{1, 2, \dots, D\}$. \square

APPENDIX B PROOF OF THEOREM 2

Proof. First, we have

$$\begin{aligned}
& \sum_{l=1}^{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{R}|} \|\mathbf{U}\mathbf{R}_j\mathbf{T}_l\|_F^2 + \|\mathbf{V}\|_F^2 \\
&= \sum_{d=1}^r \|\mathbf{u}_{:d}\|_2^2 \|\mathbf{r}_{:d}\|_2^2 \|\mathbf{t}_{:d}\|_2^2 + |\mathcal{R}| |\mathcal{T}| \|\mathbf{v}_{:d}\|_2^2, \\
& \sum_{l=1}^{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{R}|} \|\mathbf{R}_j\mathbf{T}_l\mathbf{V}^\top\|_F^2 + \|\mathbf{U}\|_F^2 \\
&= \sum_{d=1}^r \|\mathbf{v}_{:d}\|_2^2 \|\mathbf{r}_{:d}\|_2^2 \|\mathbf{t}_{:d}\|_2^2 + |\mathcal{R}| |\mathcal{T}| \|\mathbf{u}_{:d}\|_2^2.
\end{aligned}$$

Suppose that $\|\hat{\mathcal{X}}\|_*$ can be attained at D and $(\mathbf{u}_{:d}^*, \mathbf{r}_{:d}^*, \mathbf{v}_{:d}^*, \mathbf{t}_{:d}^*)$ for $d = 1, \dots, D$. Let

$$\begin{aligned}
\mathbf{u}'_{:d} &= \alpha_u \mathbf{u}_{:d}^* = \left(\sqrt{\frac{\|\mathbf{v}_{:d}^*\|_2 \|\mathbf{r}_{:d}^*\|_2 \|\mathbf{t}_{:d}^*\|_2}{\sqrt{|\mathcal{R}|} |\mathcal{T}| \|\mathbf{u}_{:d}^*\|_2}}} \right) \mathbf{u}_{:d}^* \\
\mathbf{v}'_{:d} &= \alpha_v \mathbf{v}_{:d}^* = \left(\sqrt{\frac{\|\mathbf{u}_{:d}^*\|_2 \|\mathbf{r}_{:d}^*\|_2 \|\mathbf{t}_{:d}^*\|_2}{\sqrt{|\mathcal{R}|} |\mathcal{T}| \|\mathbf{v}_{:d}^*\|_2}}} \right) \mathbf{v}_{:d}^* \\
\mathbf{r}'_{:d} &= \alpha_r \mathbf{r}_{:d}^* = \left(\sqrt{\frac{\sqrt{|\mathcal{R}|} |\mathcal{T}|}{\|\mathbf{r}_{:d}^*\|_2 \|\mathbf{t}_{:d}^*\|_2}} \right) \mathbf{r}_{:d}^* \\
\mathbf{t}'_{:d} &= \alpha_t \mathbf{t}_{:d}^* = \left(\sqrt{\frac{\sqrt{|\mathcal{R}|} |\mathcal{T}|}{\|\mathbf{r}_{:d}^*\|_2 \|\mathbf{t}_{:d}^*\|_2}} \right) \mathbf{t}_{:d}^*.
\end{aligned}$$

Notice that

$$\begin{aligned}
\hat{\mathcal{X}} &= \sum_{d=1}^D \mathbf{u}'_{:d} \otimes \mathbf{r}'_{:d} \otimes \mathbf{v}'_{:d} \otimes \mathbf{t}'_{:d}, \\
\|\hat{\mathcal{X}}\|_* &= \sum_{d=1}^D \|\mathbf{u}'_{:d}\|_2 \|\mathbf{r}'_{:d}\|_2 \|\mathbf{v}'_{:d}\|_2 \|\mathbf{t}'_{:d}\|_2,
\end{aligned}$$

i.e., $(\mathbf{u}_{:d}^*, \mathbf{r}_{:d}^*, \mathbf{v}_{:d}^*, \mathbf{t}_{:d}^*)$ for $d = 1, \dots, D$ also attains the nuclear p -norm $\|\hat{\mathcal{X}}\|_*$.

It follows from the inequality of arithmetic and geometric means that

$$\begin{aligned}
& \frac{1}{4} \sum_{d=1}^r (\|\mathbf{u}_{:d}\|_2^2 \|\mathbf{r}_{:d}\|_2^2 \|\mathbf{t}_{:d}\|_2^2 + |\mathcal{R}| |\mathcal{T}| \|\mathbf{v}_{:d}\|_2^2) \\
&+ (\|\mathbf{v}_{:d}\|_2^2 \|\mathbf{r}_{:d}\|_2^2 \|\mathbf{t}_{:d}\|_2^2 + |\mathcal{R}| |\mathcal{T}| \|\mathbf{u}_{:d}\|_2^2) \\
&\geq \sum_{d=1}^r \sqrt{|\mathcal{R}| |\mathcal{T}|} \|\mathbf{u}_{:d}\|_2 \|\mathbf{r}_{:d}\|_2 \|\mathbf{v}_{:d}\|_2 \|\mathbf{t}_{:d}\|_2 \\
&\geq \sum_{d=1}^D \sqrt{|\mathcal{R}| |\mathcal{T}|} \|\mathbf{u}'_{:d}\|_2 \|\mathbf{r}'_{:d}\|_2 \|\mathbf{v}'_{:d}\|_2 \|\mathbf{t}'_{:d}\|_2 \\
&= \sqrt{|\mathcal{R}| |\mathcal{T}|} \|\hat{\mathcal{X}}\|_*
\end{aligned}$$

The equality holds if $(\mathbf{u}_{:d}, \mathbf{r}_{:d}, \mathbf{v}_{:d}, \mathbf{t}_{:d}) = (\mathbf{u}'_{:d}, \mathbf{r}'_{:d}, \mathbf{v}'_{:d}, \mathbf{t}'_{:d})$, i.e.,

$$\begin{aligned} & \frac{1}{4\sqrt{|\mathcal{R}||\mathcal{T}|}} \sum_{d=1}^D (\|\mathbf{u}'_{:d}\|_2^2 \|\mathbf{r}'_{:d}\|_2^2 \|\mathbf{t}'_{:d}\|_2^2 + |\mathcal{R}||\mathcal{T}|\|\mathbf{v}'_{:d}\|_2^2) \\ & + (\|\mathbf{v}'_{:d}\|_2^2 \|\mathbf{r}'_{:d}\|_2^2 \|\mathbf{t}'_{:d}\|_2^2 + |\mathcal{R}||\mathcal{T}|\|\mathbf{u}'_{:d}\|_2^2) \\ & = \sum_{d=1}^D \|\mathbf{u}'_{:d}\|_2 \|\mathbf{r}'_{:d}\|_2 \|\mathbf{v}'_{:d}\|_2 \|\mathbf{t}'_{:d}\|_2 = \|\hat{\mathcal{X}}\|_{**}. \end{aligned}$$

Therefore, the minimization of the left side is $\|\hat{\mathcal{X}}\|_{**}$. \square

APPENDIX C PROOF OF THEOREM 3

Lemma 1. *The nuclear p -norm is also derived by*

$$\begin{aligned} \|\mathcal{A}\|_{p*} &= \min \left\{ \frac{1}{D} \sum_{i=1}^r \sum_{k=1}^D \|\mathbf{u}_{k,i}\|_p^D : \right. \\ & \quad \left. \mathcal{A} = \sum_{i=1}^r \mathbf{u}_{1,i} \otimes \cdots \otimes \mathbf{u}_{D,i}, r \in \mathbb{N} \right\}, \quad (16) \end{aligned}$$

where $\mathbf{u}_{k,i} \in \mathbb{R}^{n_k}$ for $k = 1, \dots, D$, $i = 1, \dots, r$, and \otimes denotes the outer product.

Proof. Suppose that the right side of Equation (16) attains $\|\mathcal{A}\|_{p*}$ at R and $(\mathbf{u}_{k,i}^*)$ for $k = 1, \dots, D$, $i = 1, \dots, R$. Let

$$\begin{aligned} \mathbf{u}'_{k,i} &= \alpha_{k,i} \mathbf{u}_{k,i}^*, \\ \alpha_{k,i} &= \frac{\sqrt[p]{\prod_{d=1}^D \|\mathbf{u}_{d,i}^*\|_p}}{\|\mathbf{u}_{k,i}^*\|}. \end{aligned}$$

Notice that

$$\begin{aligned} \mathcal{A} &= \sum_{i=1}^R \mathbf{u}'_{1,i} \otimes \cdots \otimes \mathbf{u}'_{D,i}, \\ & \quad \sum_{i=1}^R \prod_{k=1}^D \|\mathbf{u}'_{k,i}\|_p = \|\mathcal{A}\|_{p*}, \end{aligned}$$

i.e., $(\mathbf{u}'_{k,i})$ for $k = 1, \dots, D$, $i = 1, \dots, R$ also attains the nuclear p -norm $\|\mathcal{A}\|_{p*}$.

It follows from the inequality of arithmetic and geometric means that

$$\begin{aligned} \frac{1}{D} \sum_{i=1}^r \sum_{k=1}^D \|\mathbf{u}_{k,i}\|_p^D &= \sum_{i=1}^r \left(\frac{1}{D} \sum_{k=1}^D \|\mathbf{u}_{k,i}\|_p^D \right) \\ &\geq \sum_{i=1}^r \left(\prod_{k=1}^D \|\mathbf{u}_{k,i}\|_p \right) \\ &\geq \sum_{i=1}^R \prod_{k=1}^D \|\mathbf{u}'_{k,i}\|_p \\ &= \|\mathcal{A}\|_{p*} \end{aligned}$$

where $\mathcal{A} = \sum_{i=1}^r \mathbf{u}_{1,i} \otimes \cdots \otimes \mathbf{u}_{D,i}$. As the left side attains the minimization $\|\mathcal{A}\|_{p*}$ at R and $(\mathbf{u}'_{k,i})$ for $k = 1, \dots, D$, $i = 1, \dots, R$, i.e.,

$$\frac{1}{D} \sum_{i=1}^r \sum_{k=1}^D \|\mathbf{u}'_{k,i}\|_p^D = \sum_{i=1}^R \prod_{k=1}^D \|\mathbf{u}'_{k,i}\|_p = \|\mathcal{A}\|_{p*}.$$

Therefore, the minimization of the left side is $\|\mathcal{A}\|_{p*}$. \square

TABLE 7: Statistics of KGC benchmark datasets.

	WN18RR	FB15k-237	YAGO3-10
#Entity	40,943	14,541	123,182
#Relation	11	237	37
#Train	86,835	272,115	1,079,040
#Valid	3,034	17,535	5,000
#Test	3,134	20,466	5,000

TABLE 8: Statistics of temporal KGC benchmark datasets.

	ICEWS14	ICEWS05-15	YAGO15k
#Entity	6,869	10,094	15,403
#Relation	230	251	51
#Timestamp	365	4017	170
#Train	72,826	368,962	110,441
#Valid	8,941	46,275	13,815
#Test	8,963	46,092	13,800

APPENDIX D DATASET STATISTICS

For static knowledge graph completion, we consider three public static knowledge graph datasets—WN18RR [40], FB15k-237 [18], and YAGO3-10 [7], which have been divided into training, validation, and testing set in previous works. The statistics of these datasets are shown in Table 7.

For temporal knowledge graph completion, we consider three public temporal knowledge graph datasets—ICEWS14 [50], ICEWS05-15 [50], and YAGO15k [51], which have been divided into training, validation, and testing set in previous works. ICEWS14 and ICEWS05-15 are extracted from the Integrated Conflict Early Warning System (ICEWS). YAGO15k is a modification of FB15k. The statistics of these datasets are shown in Table 8.

APPENDIX E EVALUATION METRICS

The mean reciprocal rank is the average of the reciprocal ranks of results for a sample of queries Q :

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}.$$

The Hits@N is the ratio of ranks that no more than N :

$$\text{Hits}@N = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \mathbb{1}_{x \leq N}(\text{rank}_i),$$

where $\mathbb{1}_{x \leq N}(\text{rank}_i) = 1$ if $\text{rank}_i \leq N$ or otherwise $\mathbb{1}_{x \leq N}(\text{rank}_i) = 0$.

APPENDIX F THE OPTIMAL VALUE OF P IN STATIC KGC

In distance-based models, the commonly used p is either 1 or 2. When $p = 2$, DURA takes the form as the one in Equation (7) in the main text. If $p = 1$, we cannot expand the squared score function of the associated DB models as in Equation (4). Thus, the induced regularizer takes the form of $\sum_{(u_i, r_j, v_k) \in \mathcal{S}} \|\mathbf{u}_i \bar{\mathbf{R}}_j - \mathbf{v}_k\|_1 + \|\mathbf{v}_k \bar{\mathbf{R}}_j^\top - \mathbf{u}_i\|_1$. The above regularizer with $p = 1$ (Reg_p1) does not give an upper

TABLE 9: The queries in T-SNE visualizations.

Index	Query
1	(political drama, /media_common/netflix_genre/titles,?)
2	(Academy Award for Best Original Song, /award/award_category/winners./award/award_honor/ceremony,?)
3	(Germany, /location/location/contains,?)
4	(doctoral degree , /education/educational_degree/people_with_this_degree./education/education/major_field_of_study,?)
5	(broccoli, /food/food/nutrients./food/nutrition_fact/nutrient,?)
6	(shooting sport, /olympics/olympic_sport/athletes./olympics/olympic_athlete_affiliation/country,?)
7	(synthpop, /music/genre/artists,?)
8	(Italian American, /people/ethnicity/people,?)
9	(organ, /music/performance_role/track_performances./music/track_contribution/role,?)
10	(funk, /music/genre/artists,?)

bound on the tensor nuclear-2 norm as in Theorem 1. Table 10 shows that, DURA significantly outperforms Reg_p1 on WN18RR and FB15k-237. Therefore, we choose $p = 2$.

TABLE 10: Comparison to Reg_p1, where "R" denotes RESCAL and "C" denotes ComplEx.

	WN18RR			FB15k-237		
	MRR	H@1	H@10	MRR	H@1	H@10
R-Reg_p1	.281	.220	.394	.310	.228	.338
C-Reg_p1	.409	.393	.439	.316	.229	.487
R-DURA	.498	.455	.577	.368	.276	.550
C-DURA	.491	.449	.571	.371	.276	.560

APPENDIX G THE QUERIES IN T-SNE VISUALIZATION

In Table 9, we list the ten queries used in the T-SNE visualization (Section 6.5 in the main text). Note that a query is represented as $(u, r, ?)$, where u denotes the head entity and r denotes the relation.