Journal of Intelligent Information Systems (2019) 53:167–197
https://doi.org/10.1007/s10844-018-0535-2

AWML: adaptive weighted margin learning for knowledge graph embedding

Chenchen Guo[1] · Chunhong Zhang[1] · Xiao Han[1] · Yang Ji[1]

Received: 18 March 2018 / Revised: 10 October 2018 / Accepted: 11 October 2018 /
Published online: 26 October 2018

Abstract

Knowledge representation learning (KRL), exploited by various applications such as question answering and information retrieval, aims to embed the entities and relations contained by the knowledge graph into points of a vector space such that the semantic and structure information of the graph is well preserved in the representing space. However, the previous works mainly learned the embedding representations by treating each entity and relation equally which tends to ignore the inherent imbalance and heterogeneous properties existing in knowledge graph. By visualizing the representation results obtained from classic algorithm TransE in detail, we reveal the disadvantages caused by this homogeneous learning strategy and gain insight of designing policy for the homogeneous representation learning. In this paper, we propose a novel margin-based pairwise representation learning framework to be incorporated into many KRL approaches, with the method of introducing adaptivity according to the degree of knowledge heterogeneity. More specially, an adaptive margin appropriate to separate the real samples from fake samples in the embedding space is first proposed based on the sample's distribution density, and then an adaptive weight is suggested to explicitly address the trade-off between the different contributions coming from the real and fake samples respectively. The experiments show that our Adaptive Weighted Margin Learning (AWML) framework can help the previous work achieve a better performance on real-world Knowledge Graphs *Freebase* and *WordNet* in the tasks of both link prediction and triplet classification.

Keywords Knowledge graph · Knowledge representation learning · Adaptive margin · Adaptive importance weight

✉ Chunhong Zhang
zhangch@bupt.edu.cn

Extended author information available on the last page of the article.

1 Introduction

A *Knowledge Graph* (KG), which organizes human knowledge into a structured knowledge system, such as WordNet (Miller 1995) and Freebase (Bollacker et al. 2008), is a powerful database applied in knowledge inference (Minervini et al. 2016; Zhang et al. 2017; Han et al. 2018), information retrieval (Metzger et al. 2017), question answering (Ferrández et al. 2016), and many other fields, promoting the development of artificial intelligence. A knowledge fact existing in KG is denoted as a discrete triple $\langle h, r, t\rangle$ where h, r, t indicate a head entity, a relation, and a tail entity, respectively. For example, in Freebase, a triple $\langle$ Steve Jobs, *PlaceOfBirth*, San Francisco $\rangle$ indicates a fact that the person Steve Jobs was born in the place San Francisco, where Steve Jobs is the head h of a triple, San Francisco is a tail t and *PlaceOfBirth* is a relation r.

As the size of KG increases and the computation complexity arises from the heterogeneity and sparsity of knowledge graph, *Knowledge Representation Learning* (KRL) has been attracting massive research attention to project semantically similar points from the data manifold in KG onto metrically close points in a low-dimensional embedding space. Analogously, different points in KG should be projected onto metrically distant points in the embedding space. Particularly, the three components in the triple $\langle h, r, t\rangle$ are encoded as three embeddings $\boldsymbol{h}$, $\boldsymbol{r}$ and $\boldsymbol{t}$ respectively. The rationality of the embedded $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}$ is evaluated by a semantic score function f that measures point distances in the embedding space. For example in TransE (Bordes et al. 2013), the score function is chosen as $f(h, r, t) = \|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|_{L_n}$, which indicates $\boldsymbol{t}$ should be close to $\boldsymbol{h} + \boldsymbol{r}$ in the metric of L_n distance. In other words, the smaller the distance between the relation $\boldsymbol{r}$ and the difference of two entities $\boldsymbol{t} - \boldsymbol{h}$, the higher the confidence of a triple is held and therefore the better the KG is preserved.

To facilitate the triples of KG to obtain the optimal scores during the preservation, in addition to the explicit triples, the synthetic fake triples obtained by "negative sampling (see (2))" from KG are also involved. Triplet loss then demands that the difference of the distance scores between the reals and the fakes be larger than some pre-assigned margin constant. So in contrast to the real triples, the score of the fake triples should be enlarged to distinguish them from the real ones. This training strategy based on both real and fake triples forms the training objective of the KRL model and is commonly called margin-based pairwise learning algorithm (Jenatton et al. 2012; Bordes et al. 2014; Zhou et al. 2016), where the constant margin is selected as the hyper-parameter of the model to separate the real score and the fake score.

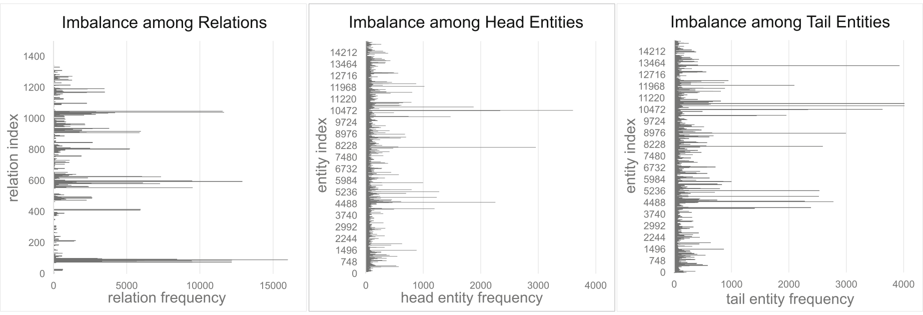

Fig. 1 Imbalance in FB15k KG. In the *Left*, *Middle* and *Right*, the length of each column represents the existing frequency in KG for each relation, head entity or tail entity, respectively

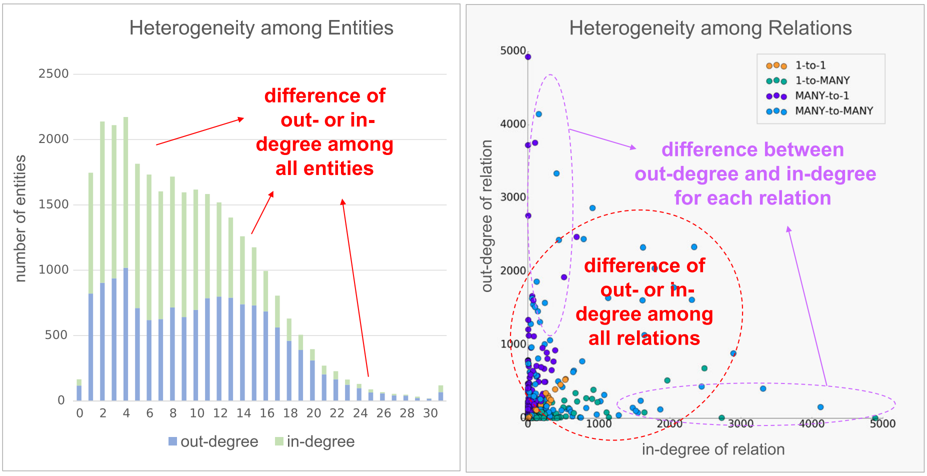

Fig. 2 Heterogeneity in FB15k KG. *Left:* The length of each column represents how many entities have the corresponding out-degree nr_h or in-degree nr_t. *Right*: Each circle represents a relation. Its coordinate depends on the in-degree nh_r and out-degree nt_r of the relation, and its color depends on the type of relation: 1-to-1, 1-to-MANY, MANY-to-1 and MANY-to-MANY

However, that simple constant strategy of margin selection indicates a fixed boundary between the reals and the fakes, which is obviously inconsistent with the complex properties of KG — the imbalance and heterogeneity as shown in Figs. 1 and 2. The imbalance property refers to the fact that each relation occurs in KG many times and the occurring frequency differs from relation to relation, and so as the entity. The heterogeneity is considered as 6 kinds of difference: a) the difference of out-degree nr_h among all entities; b) the difference of in-degree nr_t among all entities; c) the difference of out-degree nh_r among all relations; d) the difference of in-degree nt_r among all relations; e) the difference between out-degree nh_r and in-degree nt_r for a relation, f) the difference of hpt_r and tph_r among all relations,[1] where these arguments are denoted in Table 1. The above properties of imbalance and heterogeneity imply different knowledge categories — the triples $\langle h, r, t\rangle$ in KG can be categorized into different types in terms of the imbalance and heterogeneity of either entity h/t or the relation r. Through visualization of embeddings elaborated in Section 3, we discover that the diversity among knowledge categories will bring about diversity among the distribution density of embedding points during the KG preservation. Then, the fixed separating margin is no longer in the same order of magnitude with each category-specific density. The previous homogeneous learning strategy is no longer appropriate for the representation of the imbalanced and heterogeneous KG. Therefore, the separating margin in the original learning algorithm should be adjusted adaptively according to the category-specific density to facilitate the preservation of KG.

Furthermore, the optimization of the real-triple score and the fake-triple score is of equal importance in the previous margin-based pairwise learning algorithm. However, through visualization, we find that for different knowledge categories, either the real-triple score function f or the fake-triple one is under-restricted in different degree. Thus, the trade-off

[1] We compute hpt_r and tph_r to classify the relations into 4 types: 1-to-1, 1-to-MANY, MANY-to-1 and MANY-to-MANY, following Bordes et al. (2013). If the average number hpt_r or tph_r is below 1.5 then the argument is labeled as 1 and MANY otherwise.

Table 1 Denotations

nr_h	The number of relations for a head
nr_t	The number of relations for a tail
nh_r	The number of heads for a relation
nt_r	The number of tails for a relation
hpt_r	The averaged number of heads per tail for a relation
tph_r	The averaged number of tails per head for a relation

between the contributions coming from the real and fake triples should be controlled in different degree depending on the category of knowledge.

Though many improvements have involved redesigning or modifying the basic framework with regard to the semantic measurement of score function f such as KG2E (He et al. 2015), ProjE (Shi and Weninger 2017), etc, the underlying training objective is rarely concerned in literature. Therefore, in this work, we emphasize the high-level objective independent of the concrete form of f. With the method of introducing the concept of density-adaptive margin and density-adaptive weight into the previous margin-based pairwise framework, we propose an *Adaptive Weighted Margin Learning* AWML algorithm which can be potentially incorporated into many existing KRL approaches regardless of the complexity. Besides, we also disambiguate the relations to make the model perform more precisely. In our visualization analysis and experiment, two typical real-world KGs, Freebase and WordNet, are selected to build datasets and carry out evaluation on two tasks, including link prediction and triplet classification. Experimental and visualized results demonstrate that our general AWML algorithm can significantly improve the performance of KRL models and result in a more expressive representation.

Contributions The main contributions of this work are concluded as follows:

- Through visualization analysis, we explore the category-specific distributed density and discover the inconsistency between the original training objective and the complex property of KG.
- We retrofit the original margin-based pairwise algorithm and propose a novel one by adding the adaptive weight and the adaptive margin into the training objective. The experiment and visualization of AWML both demonstrate its capability of equilibrating all the knowledge category and controlling the trade-off between the real and fake triples.
- We evaluate our retrofitted algorithm, AWML, in the tasks of link prediction and triplet classification. The results show empirically that our adaptive methods end up being powerful on such applications.

Outline In this work, we propose an adaptive framework appropriate for the KRL models. Two adaptive methods are utilized to solve the limits of KRL models. After exploring the representation distribution and the spatial density, we propose a density-adaptive margin and a density-adaptive weight in the training objective of KRL models. The evaluation results on Freebase and WordNet KGs indicates that our proposed framework has the capability to help the KRL model to achieve the better embeddings in the representation space.

The rest of the paper is organized as follows. In Section 2, we introduce the original margin-based pairwise criterion and the existing KRL models. In Section 3 we visualize the spatial distribution characteristics of embedding representations and discover two limitations of previous KRL models: a) the inflexibility over importance trade-off, b) the inflexibility over separating margin. Then in Section 4, we introduce the density-adaptive importance weight and the density-adaptive margin to propose a novel framework, AWML, to be incorporated into the previous KRL models, while in Section 5 we empirically evaluate the proposed learning framework. In Section 6, we discuss our proposed work and analysis method. Finally, we summarize our work and outline future research directions.

2 Related work

In this section, we review the origin of margin-based training objective and how the existing KRL models utilize it. Then, we summarize the triplet score function f over different classical KRL models. Note that our AWML framework is independent of the concrete form of score function f, and so, can be potentially incorporated into all KRL models.

2.1 Margin-based pairwise learning criterion

The notion of margin is generalized by a commonly-used classifier SVM (Weston and Watkins 1999; Boser et al. 1992), which maximizes the margin value between the training patterns and the decision boundary so that two classes can be separated in the feature space as precise as possible. In order to suit for multi-classification, some researchers extend SVM and introduce the margin-based pairwise learning criterion to take all the classes into account simultaneously. Such form of margin-based pairwise objective has been also applied in knowledge representation to separate the reals and the fakes in the embedding space.

The training objective of a distance-based KRL model is typically to minimize the following margin-based pairwise function:

$$L(S) = \sum_{\langle h,r,t\rangle \in S} \sum_{\langle h',r',t'\rangle \in S'_{\langle h,r,t\rangle}} [\gamma + f(h,r,t) - f(h',r',t')]_+, \tag{1}$$

where the real-triple score $f(h,r,t)$ and the fake-triple score $f(h',r',t')$ are measured simultaneously. The score function f represents the semantic similarity of a triple, i.e. the probability of a triple to be true. For distance-based KRL models, the score function $f(h,r,t)$ is designed as some distance restriction among three components h,r,t in the triple.

To minimize the training objective is not only to get a real triple score $f(h,r,t)$ lower than all the corresponding fake triple score $f(h',r',t')$, but also to make the difference between such two kinds of triple scores at least higher than a positive constant, the margin γ. The fake triple is sampled by randomly replacing the head, the tail, or the relation of a real triple. The replacement rule as follows:

$$S'_{\langle h,r,t\rangle} = \{\langle h',r,t\rangle | h' \in E\} \cup \{\langle h,r',t\rangle | r' \in R\} \cup \{\langle h,r,t'\rangle | t' \in E\}, \tag{2}$$

where E and R refer to the entity set and the relation set in KG respectively.

2.2 Existing KRL models

Different KRL models formulate their score function $f(h,r,t)$ based on different designs of semantic similarity measurement, which further lead to various training objectives. In this subsection, we summarize some KRL models and their distinctive similarity measurement of a triple.

Translation-based embedding methods Inspired by the translation-invariant phenomenon of word embeddings in the work of word2vec (Mikolov et al. 2013), **TransE** (Bordes et al. 2013) model regards a relation as an embedding vector $\boldsymbol{r}$ that indicates the semantic translation from the head entity $\boldsymbol{h}$ to the tail entity $\boldsymbol{t}$ for each real triple $\langle h, r, t\rangle$. In order to satisfy the approximation $\boldsymbol{h} + \boldsymbol{r} \approx \boldsymbol{t}$ when the triple $\langle h, r, t\rangle$ holds, the score function of a triple is designed as $\|\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}\|_{L_n}$, measuring the L_n-distance between a translated head entity $\boldsymbol{h} + \boldsymbol{r}$ and some tail entity $\boldsymbol{t}$.

Compared to traditional methods, TransE model can well balance the effectiveness and computational cost, while the over-simplified translation assumption encounters a challenge when dealing with complicated relations including 1-to-MANY, MANY-to-1, and MANY-to-MANY relations (Bordes et al. 2013). In order to solve this problem, **TransH** (Wang et al. 2014), **TransR** (Lin et al. 2015b) and **TransD** (Ji et al. 2015) translate embeddings based on relation-specific hyperplanes, relation-specific entity projection and relation-specific dynamic mapping respectively. However, in TransR, simple relations may be overfitting or complex relation may be underfitting because every relation (no matter complex or simple) has the same number of parameters to learn. **KG2E** (He et al. 2015) and **TransG** (Xiao et al. 2016) attempt to retrofit the model with the Gaussian probability distribution. KG2E performs relatively well on 1-to-N and N-to-1 relations. Furthermore, some KRL model enhance translation-based model with other information in addition to triple-based semantic information inherent in the graph structure, and for instance, **PTransE** (Lin et al. 2015a) utilizes path information between two entities and **DKRL** (Xie et al. 2016) utilizes entity description.

Other embedding methods In addition to translation-based models, there are also many other embedding methods following the margin-based pairwise learning criterion. We list the seven typical models here and most of their score function are listed in Table 2 respectively. Their parameters corresponding to the relation are also displayed in the last column of Table 2. Note here that in Table 2, the $\boldsymbol{M_r}$ denotes a transformation matrix specific for the relation r. The $\boldsymbol{h}$, $\boldsymbol{r}$ and $\boldsymbol{t}$ indicate the embedding vector of the head h, the relation r and the tail t.

SE model (Bordes et al. 2009) designs two independent relation-specific projections for head and tail entities and then compute their distance. **SME** model (Bordes et al. 2014; 2012) encodes not only each entity but also each relation into a vector and utilizes linear algebra operations in a neural network to capture correlations between entities and relations.

Table 2 Scoring functions on triplet $\langle h, r, t\rangle$ of different KRL models, and their relation-dependent parameters

Model	Score function $f(h, r, t)$	Relation parameters
SE	$\\|\boldsymbol{M}_{rh}\boldsymbol{h}-\boldsymbol{M}_{rt}\boldsymbol{t}\\|_{L_1}$	$\boldsymbol{M}_{rh}, \boldsymbol{M}_{rt} \in \mathbb{R}^{d\times d}$
SME(linear)	$(\boldsymbol{M}_{r1}\boldsymbol{h}+\boldsymbol{M}_{r2}\boldsymbol{r}+\boldsymbol{b}_r)^\top(\boldsymbol{M}_{r1}\boldsymbol{t}+\boldsymbol{M}_{r2}\boldsymbol{r}+\boldsymbol{b}_r)$	$\boldsymbol{M}_{r1}, \boldsymbol{W}_{r2} \in \mathbb{R}^{k\times d}, \boldsymbol{b}_r \in \mathbb{R}^{k\times 1}$
SME(bilinear)	$(\boldsymbol{M}_{r1}\boldsymbol{h} \bigotimes \boldsymbol{M}_{r2}\boldsymbol{r}+\boldsymbol{b}_r)^\top(\boldsymbol{M}_{r1}\boldsymbol{t} \bigotimes \boldsymbol{M}_{r2}\boldsymbol{r}+\boldsymbol{b}_r)$	$\boldsymbol{M}_{r1}, \boldsymbol{M}_{r2} \in \mathbb{R}^{k\times d}, \boldsymbol{b}_r \in \mathbb{R}^{k\times 1}$
NTN	$\boldsymbol{r}^\top tanh(\boldsymbol{h}\boldsymbol{M}_r\boldsymbol{t}+\boldsymbol{M}_{r,1}\boldsymbol{h}+\boldsymbol{M}_{r,2}\boldsymbol{t}+\boldsymbol{b}_r)$	$\boldsymbol{M}_r \in \mathbb{R}^{d\times d\times k}, \boldsymbol{M}_{r,1}, \boldsymbol{M}_{r,2} \in \mathbb{R}^{k\times d}, \boldsymbol{b}_r \in \mathbb{R}^{k\times 1}$
RESCAL	$\boldsymbol{h}^\top\boldsymbol{M}_r\boldsymbol{t}$	$\boldsymbol{M}_r \in \mathbb{R}^{d\times d}$
Hole	$\sigma(\boldsymbol{r}^\top(\boldsymbol{h} * \boldsymbol{t}))$	–

NTN (Socher et al. 2013) considers the second-order correlations into nonlinear neural networks. **ProjE** model (Shi and Weninger 2017) utilizes combination operation and nonlinear transformations based on neural networks, while Zhao et al. (2017) uses convolutional neural network (**CNN**) to learn the sequential entity and relation representations. **RESCAL** model (Nickel and Ring 2012; Nickel et al. 2011) utilizes matrix factorization with every value of a three-dimensional tensor, where the value of 1 for real triples and 0 for fake triples will be all factorized approximately into the form of $\boldsymbol{h}^{\top}\boldsymbol{M}_r\boldsymbol{t}$. **Hole** model (Nickel et al. 2015) introduces an operation of circular correlation $*$ between head and tail to represent this entity pair so that every dimension of the entity embedding is correlated with other dimensions: $[\boldsymbol{h} * \boldsymbol{t}]_k = \sum_{i=0}^{d-1} [\boldsymbol{h}_i \boldsymbol{t}_{(i+k) \bmod d}]$.

All these KRL models modify or redesign the semantic measurement of f based on the margin-based pairwise training objective (1). However, such form of training objective neglects the complex property of KG and treats all the knowledge categories equally without discrimination, which limits the performance of knowledge representation.

3 Objective analysis with visualization

To look deep into the limitation of the embedding properties of the previous works, in this section we analyze the representations by visualizing the embedding space. We take the **TransE** as an analysis example for simplicity and the results could be smoothly extended to other models with the similar underlying principles with **TransE**. In particular, for a triple $\langle h, r, t\rangle$, the embedding vectors of the three elements are transformed by t-SNE (Maaten and Hinton 2008) into the coordinates of three points in a 2D plane. By plotting a set of triples coming from different relation categories, we observe the distribution patterns and the densities of embedding points, based on which the shortcomings of distance-based KRL models are revealed and then the insight of our algorithm improvement is gained.

In the following, we start by introducing our analysis approach to explore the distribution pattern of the representations. Please note that here, the representation we observe is the implicit embedding vector of each triple. Take TransE model as an example, for the triple $\langle h, r, t\rangle$, we take $\boldsymbol{t} - \boldsymbol{h}$ as the implicit embedding vector of the triple and visualize it in our observation. And then, we display the phenomenon of relational semantic diversity. As for this problem, we give out our solution to it to make the model perform more precisely and take it as the prerequisite of our algorithm. Afterwards, through the exploration of distribution density, we explain our idea of adaptivity on the basis of two kinds of inflexibility over the previous distance-based KRL models: inflexibility over **importance trade-off** and inflexibility over **separating margin**.

3.1 Representation distribution and semantic diversity

Representation distribution observation Structured in the form of a graph, entities, and relations are projected into a continuous embedding space by some specific measurement of semantic similarity, which makes the embedding space fitted into the semantic space. With the goal of exploring such a structure of the embedding space and analyzing the performance of KRL model further, we visualize the knowledge embeddings with the help of a dimensionality reduction technique, t-SNE (Maaten and Hinton 2008). Such a dimensionality reduction technique will highly match and display the graph position and the structure of their local graph neighborhoods in the distributed embedding space.

In this paper, we take TransE model as a proof of principle and take a typical real-world dataset, FB15k (Bollacker et al. 2008) as a visualization dataset whose statistics and peculiar characteristics are listed in Section 5. As for the triple-wise measurement of score function f, TransE interprets the distance between vectors of $\boldsymbol{t} - \boldsymbol{h}$ and $\boldsymbol{r}$ as the semantic similarity of an triple $\langle h, r, t \rangle$. Once the training objective of TransE is optimized over the whole KG for long enough, all the implicit vectors $\boldsymbol{t} - \boldsymbol{h}$ of the real triples with the same relation will eventually form a single cluster near the relation $\boldsymbol{r}$ in the embedding space, and they are not required to collapse to a single point; they merely need to be closer to each other than to any offset with a different relation.

Thus, we take training triples $\langle h, r, t \rangle$ with the same relation r as a category of knowledge, i.e., an observed collection. Then, the training set S is divided into multiple triple

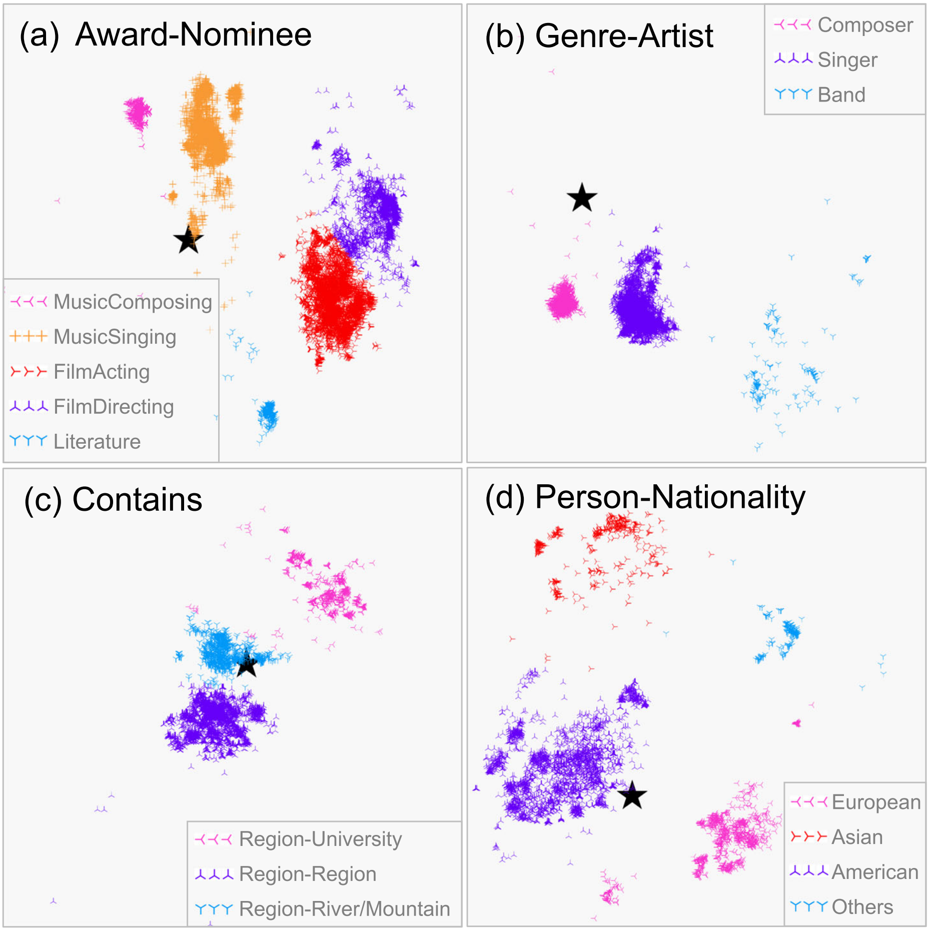

Fig. 3 Visualization results of TransE embedding vectors with t-SNE dimension reduction. Four relations ($a \sim d$) are chosen from FB15k. A black star denotes each relation embedding $\boldsymbol{r}$, and a colorful dot denotes the entity-pair offset $\boldsymbol{t} - \boldsymbol{h}$ of each golden triple. Different colors or symbols represent different latent semantics of a specific relation

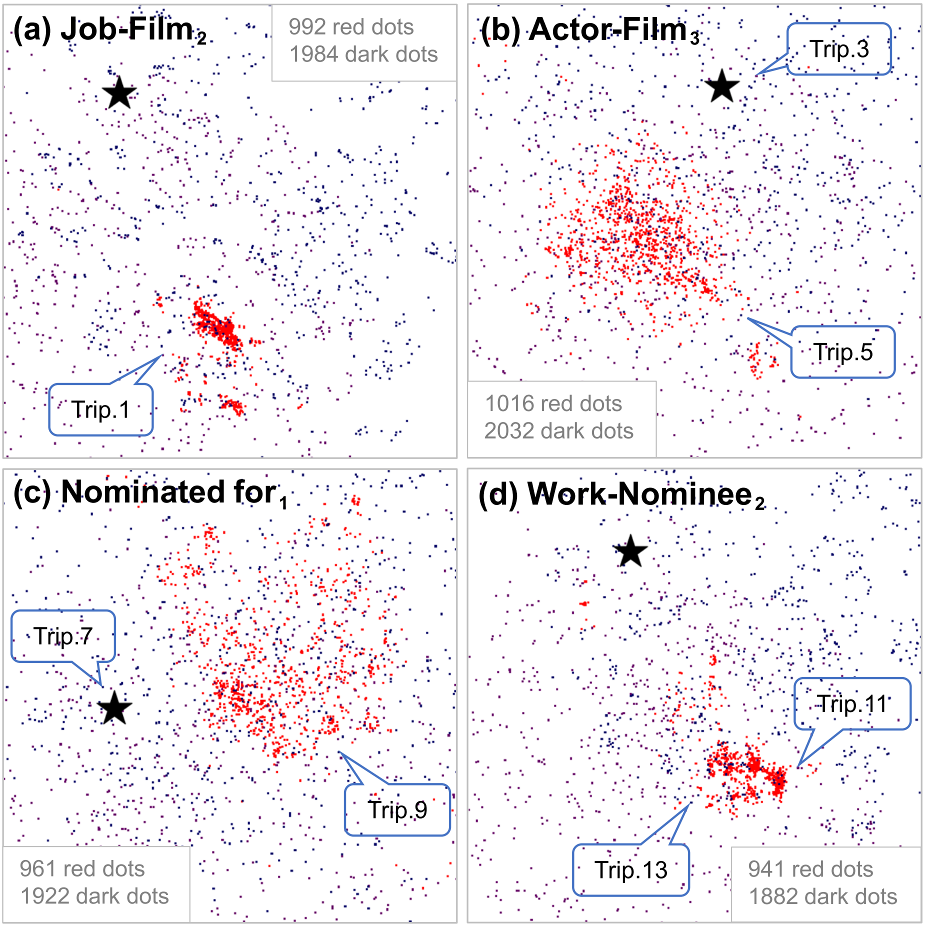

Fig. 4 Visualization results of CTransE embedding vectors with t-SNE dimension reduction. Four relations ($a \sim d$) are chosen from clustered relation set R_c that contains 2291 relations, and each clustered relation is denoted in the form of $Relation_N$. Each graph is visualized in the same size in the 2D plane. A black star denotes each relation embedding $\boldsymbol{r}$ and a colorful dot denotes the entity-pair offset $\boldsymbol{t} - \boldsymbol{h}$ of each triple as shown in the legend: **red dot** represents golden triple and **dark dot** represents synthetic triple. Some concrete synthetic triples are marked in each graphs, whose semantics are shown in Table 3

categories S_r: $S = \{S_r | r \in R\}$. In Figs. 3 and 4, we visualize the specific relation $\boldsymbol{r}$ and all the entity-pair offsets $\boldsymbol{t} - \boldsymbol{h}$ for each category S_r. To explore whether the common margin-based learning criterion is capable of capturing the complex interactive patterns between entities and relations, we observe the spatial distribution to analyze whether it matches the triplet semantic. Furthermore, to analyze the **pairwise** objective, we visualize not only the golden entity-pair offsets but also the synthetic ones in Fig. 4.

Please note that here, the **golden** entity-pair refers to the pair of head and tail entities in the real triple, i.e. $\langle h, t\rangle$ in the real-life triple $\langle h, r, t\rangle$. The **synthetic** entity-pair refers to the entity-pair in the fake triple, i.e. $\langle h', t\rangle$ or $\langle h, t'\rangle$ where the h' and t' is randomly corrupted by another entity in the golden triple $\langle h, r, t\rangle$.

Relational semantic diversity First, we are interested in the restriction of TransE caused by relation semantic diversity. To do so, we visualize the embedding results of the triples on all the relations from FB15k, and randomly pick and display **4** of them in Fig. 3. As shown in Fig. 3a, the embedded relation r = *Award-Nominee* is plotted in the center as a black star. The Triples $\langle h, r, t\rangle$ containing r are also plotted. To clearly demonstrate the embedding accuracy, rather than the individual embedding vectors of h and t, we only plot the difference $\hat{\boldsymbol{r}} = \boldsymbol{t} - \boldsymbol{h}$, i.e. the synthetic-triple implicit vector, as a point in the $2D$ plane for each Triple. As commonly regarded, the closer the $\hat{\boldsymbol{r}}$ to $\boldsymbol{r}$, the more appropriate is the embedding of the Triple. However, as we can see, the embeddings of $\hat{\boldsymbol{r}}$ did not closely center around that of $\boldsymbol{r}$. In fact, they clearly present clustering characteristic, each of which we plot with different color in Fig. 3a for emphasis.

To deeply understand the underlying cause of the multi-cluster phenomenon shown on the $\hat{\boldsymbol{r}}$'s, we use Google Knowledge Graph Search API[2] to collect the semantic of entity-pair in each Triple to obtain the relation semantic. Then, we discover that different clusters represent different latent semantics, which is shown in the legend of each visualization result of Fig. 3.

As shown in Fig. 3a, the relation *Award-Nominee* has five latent semantics : *MusicComposing*-related, *MusicSinging*-related, *FilmActing*-related, *FilmDirecting*-related and *Literature*-related, and some Triples are exampled in Table 3. For instance, the *FilmDirecting*-related latent semantic of Triple ⟨ Academy Award for Best Film Editing, *Award-Nominee*, Robert Wise(a film director) ⟩ is dependent on its entity pair ⟨ Academy Award for Best Film Editing, Robert Wise(a film director) ⟩, while the *Literature*-related latent semantic of Triple ⟨ Nobel Prize in Literature, *Award-Nominee*, Thomas Mann (a novelist) ⟩ is dependent on its entity pair ⟨ Nobel Prize in Literature, Thomas Mann (a novelist) ⟩. Such property of relational semantic diversity in KG will lead to the distributed divergence of $\hat{\boldsymbol{r}}$ with the embedding of TransE-like models.

Therefore, it is unsuitable for TransE-like models to learn a unique embedding $\boldsymbol{r}$ for a multi-semantic relation, which may be under-representative to fit all entity-pairs under this relation. In order to better model these relations, we segment each category of triples S_r into several groups with the method of clustering following the idea of CTransR (Lin et al. 2015b). Afterwards, a separate embedding vector is obtained by the KRL model for each latent semantic. Specifically, each relation r is multi-projected into the embedding space as $\{\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_n\}$, each of which characterizes one latent semantic of the relation r, and the number n is decided by the clustering result. In the following, we will denote each multi-projected relation in the form of $Relation_N$. For instance, in Fig. 3c, we will distinguish the three clusters $Contains_1$, $Contains_2$, or $Contains_3$, where the relational semantics are automatically clustered to represent the meaning of associated entity pairs.

In the rest of this paper, we call the cluster-based TransE-like model as CTransX[3] and take CTransX as a proof of principle to conduct the following visualization and experiment. As for the total number of relations after clustering, we list the statistics for some KRL models in Section 5. Take CTransE as an example, we finally obtain 2291 relational embeddings over 1345 relations in FB15k. In other words, there are 2291 knowledge categories after clustering: $\{S_{r_1}, S_{r_2}, \cdots, S_{r_{2291}}\}$.

[2] The Knowledge Graph API lets us search Google Knowledge Graph for entities that match the constraints. This API is available at https://developers.google.com/knowledge-graph/.

[3] The X in CTransX can be replaced by E, R, etc., which refers to CTransE or CTransR respectively.

Table 3 Multiple latent semantics of the relation **Award-Nominee**

Latent semantics	Triples ⟨ Head, Relation, Tail ⟩
MusicComposing	⟨ Tony Award for Best Original Score, **Award-Nominee**, Alan Menken(a misical composer) ⟩
	⟨ Grammy Award for Best Pop Vocal Album, **Award-Nominee**, Sarah McLachlan(a musician) ⟩
MusicSinging	⟨ Grammy Award for Best Rap Performance by a Duo or Group, **Award-Nominee**, (a rapper) ⟩
	⟨ MTV Video Music Award for Best Art Direction, **Award-Nominee**, (a singer) ⟩
FilmActing	⟨ Razzie Award for Worst Supporting Actor, **Award-Nominee**, Anthony Quinn(an actor) ⟩
	⟨ Academy Award for Best Actor in a Supporting Role, **Award-Nominee**, Judd Hirsch(an actor) ⟩
	⟨ MTV Movie Award for Best Kiss, **Award-Nominee**, Shia LaBeouf(an actor) ⟩
FilmDirecting	⟨ Satellite Award for Best Adapted Screenplay, **Award-Nominee**, Pearly Gates (a film director) ⟩
	⟨ Academy Award for Best Original Screenplay, **Award-Nominee**, Brad Bird(a filmmaker) ⟩
	⟨ Academy Award for Best Film Editing, **Award-Nominee**, Robert Wise(a film director) ⟩
Literature	⟨ Nobel Prize in Literature, **Award-Nominee**, Thomas Mann (a novelist) ⟩
	⟨ Latin Grammy Award for Record of the Year, **Award-Nominee**, Ricky Martin(an author) ⟩

3.2 Inflexibility over importance trade-off

In addition to the semantic diversity, we also explore in the training objective of KRL models whether the golden triple or the synthetic triple is insufficient to be restricted. In other words, the question we consider is whether the spatial distribution of learned embeddings matches the triple restriction of TransE: $\boldsymbol{t} - \boldsymbol{h} \approx \boldsymbol{r}$. Furthermore, we also consider whether the learning/restrictions of the goldens and the synthetics are out of balance or not.

To this end, when visualize the embeddings, we consider not only the golden Triples $\langle h, r, t\rangle$ but also the synthetic Triples $\langle h', r, t\rangle$ or $\langle h, r, t'\rangle$, each of which is plotted as a point in the 2D plane. To display the distributed correlation of the goldens and the synthetics, we pick **4** typical relations to show in Fig. 4. The position of each Triple depends on its difference of tail and head: $\hat{\boldsymbol{r}} = \boldsymbol{t} - \boldsymbol{h}$ for the goldens (red dots), $\hat{\boldsymbol{r}}' = \boldsymbol{t} - \boldsymbol{h}'$ (purple dots) or $\boldsymbol{t}' - \boldsymbol{h}$ (blue dots) for the synthetics. As commonly regarded, the closer the $\hat{\boldsymbol{r}}$ to $\boldsymbol{r}$ and the further the $\hat{\boldsymbol{r}}'$ to $\boldsymbol{r}$,[4] the more appropriate is the embedding of the Triple.

Nevertheless, as can be seen from Fig. 4, for some relations such as *Job-Film*$_2$ and *Actor-Film*$_3$, there exist much deviation between the relation embedding $\boldsymbol{r}$ and the golden entity-pair cluster, which is contrary to the golden triple restriction of TransE $\boldsymbol{t} - \boldsymbol{h} \approx \boldsymbol{r}$. Consequently, we attempt to move the relation embedding $\boldsymbol{r}$ to the center of the golden cluster by making the golden triple score function $f(h, r, t)$ reach the minimum regardless of the synthetic one $f(h', r', t')$. Surprisingly, we find that, over the total 1345 categories of knowledge, there are 396 categories whose evaluation results are improved (the evaluate metric of MeanRank, which will be elaborated in Section 5), even there are 230 categories improved by 10% and 35 categories improved by 50%. This phenomenon indicates that for some categories, the **golden** triple lack of restriction in the previous work and should be paid more attention to in the training objective.

On the other hand, as shown in each graph, for some synthetic entity pairs $\langle h', t\rangle$ or $\langle h, t'\rangle$ that are semantically irrelevant with the relation r, their offset $\hat{\boldsymbol{r}}'$ are interwoven with the golden entity-pair offset $\hat{\boldsymbol{r}}$ or in the neighborhood of the relation $\boldsymbol{r}$. For instance in Fig. 4b, the synthetic entity pair of ***Trip*.3**: ⟨ FilmFlex(a company), Kung Fu Panda 2(a film) ⟩ is actually connected by the relation *Distributor-Film*$_3$ in KG, as ***Trip*.4** displayed in Table 4, but its offset $\hat{\boldsymbol{r}}'$ positions in the neighborhood of the embedding of relation *Actor-Film*$_2$ that is semantically different with the relation *Distributor-Film*$_3$. This phenomenon is also exist for other synthetic triples in Table 4. The above problem reveals the under-restriction of the **synthetic** triple for some knowledge categories.

Consequently, we can say that for any category, the under-restriction of either the golden triples or the synthetic triples exists in the previous KRL models. Hence, for some categories of triples, their importance in the training objective should be finetuned. So in the previous KRL models, it is inflexible for the trade-off between the goldens' importance and the synthetics' importance. This is exactly the reason why we name this section as the "Inflexibility over importance trade-off". Based on this problem of inflexibility, we should control the contributions of these two restrictions: $f(h, r, t)$ and $f(h', r', t')$ flexibly.

In the work of Miyamoto and Cho (2016), a gate is utilized to combine word-level and character-level representations. Moreover, another work Yang et al. (2016) improve the gating mechanism by using an adaptive gate to adaptively find the optimal mixture of those

[4] It is in a sense of average that the $\hat{\boldsymbol{r}}$ should be close to the $\boldsymbol{r}$ and that the $\hat{\boldsymbol{r}}'$ should be further away from $\boldsymbol{r}$. And for the synthetics $\hat{\boldsymbol{r}}'$, being further away from $\boldsymbol{r}$ is relative and is compared to the goldens $\hat{\boldsymbol{r}}$.

Table 4 Triple examples in Fig. 4

Golden/Synthetic	Triples ⟨ Head , **Relation** , Tail ⟩
Synthetic $\boldsymbol{Trip}$.1	⟨ Sony Pictures Classics(a company) , **Job-Film**$_2$, The Imaginarium of Doctor Parnassus(a film) ⟩
Golden $\boldsymbol{Trip}$.2	⟨ Sony Pictures Classics(a company) , **Distributor-Film**$_1$, The Imaginarium of Doctor Parnassus(a film) ⟩
Synthetic $\boldsymbol{Trip}$.3	⟨ FilmFlex(a company) , **Actor-Film**$_3$, Kung Fu Panda 2(a film) ⟩
Golden $\boldsymbol{Trip}$.4	⟨ FilmFlex(a company) , **Distributor-Film**$_2$, Kung Fu Panda 2(a film) ⟩
Synthetic $\boldsymbol{Trip}$.5	⟨ Om Puri(an actor) , **Actor-Film**$_3$, Filmfare Award for Best Supporting Actor ⟩
Golden $\boldsymbol{Trip}$.6	⟨ Om Puri(an actor) , **Nominee-Award**$_2$, Filmfare Award for Best Supporting Actor ⟩
Synthetic $\boldsymbol{Trip}$.7	⟨ Academy Award for Best Director , **Nominated for**$_1$, Sydney Pollack(a director) ⟩
Golden $\boldsymbol{Trip}$.8	⟨ Academy Award for Best Director , **Award-Winner**$_3$, Sydney Pollack(a director) ⟩
Synthetic $\boldsymbol{Trip}$.9	⟨ Academy Award for Best Writing Adapted Screenplay, **Nominated for**$_1$, Robert Towne(a scriptwriter) ⟩
Golden $\boldsymbol{Trip}$.10	⟨ Academy Award for Best Writing Adapted Screenplay , **Award-Nominee**$_4$, Robert Towne(a scriptwriter) ⟩
Synthetic $\boldsymbol{Trip}$.11	⟨ Actor , **Work-Nominee**$_2$, Madonna ⟩
Golden $\boldsymbol{Trip}$.12	⟨ Actor , **Profession-People**$_1$, Madonna ⟩
Synthetic $\boldsymbol{Trip}$.13	⟨ Skyfall(a film) , **Work-Nominee**$_2$, Academy Award for Best Sound Mixing ⟩
Golden $\boldsymbol{Trip}$.14	⟨ Skyfall(a film) , **Work-Award**$_3$, Academy Award for Best Sound Mixing ⟩

The synthetic head h' is denoted as purple, and the synthetic tail t' is denoted as blue

two inputs. Inspired by these two works, we adopt an adaptive weight to control the contributions coming from the goldens and the synthetics in our proposed framework. The details of our framework are elaborated in Section 4.

3.3 Inflexibility over separating margin

In the above subsection, we discover the inflexibility over importance trade-off between the golden restriction and the synthetic restriction in the KRL training objective. With the same visualizing method, in this subsection, we explore the spatial density of the embedding distribution through visualization. Note that, in our work, the spatial density indicates whether the embedding dots distribute densely or sparsely in the representation space.

From Fig. 4, we can discover that for each relation, the spatial density of golden entity-pair cluster (red dots) is various from one another. For instance, the golden cluster of relation *Job-Film*$_2$ has a higher density than that of relation *Actor-Film*$_3$, even though they have the similar number of golden triples: 992 and 1016 respectively. This phenomenon derives from the property of heterogeneity and imbalance existing in KG.

Take TransE as an example, though every triple is restricted by the approximation $\boldsymbol{t}-\boldsymbol{h} \approx \boldsymbol{r}$, if there exist too many entity-pairs $\langle h, t\rangle$ connected with the identical relation r in KG, the corresponding offsets $\hat{\boldsymbol{r}} = \boldsymbol{t} - \boldsymbol{h}$ may be projected into relatively discrete positions in the embedding space. This is because there is insufficient space in the neighborhood of the relation embedding $\boldsymbol{r}$ to accommodate too many embedded entity-pair offsets $\hat{\boldsymbol{r}}$. Take 1-to-MANY relation as another example, for triples $\langle h, r, t\rangle$ with the same relation r and the same head entity h, if the semantics of their tail entity t are totally distinctive, these tail entities will be projected into distinctive positions. Therefore, the **spatial density** of clusters of entity-pair offsets $\hat{\boldsymbol{r}}$ will vary from relation to relation following the **occurrence frequency** of the corresponding connected relations r as shown in Fig. 4, and so as the density of head $\boldsymbol{h}$ or tail $\boldsymbol{t}$ clusters not shown in this paper.

How occurrence frequency affects spatial density? In order to explore the correlation between **occurrence frequency** and **spatial density**, for each knowledge category S_r, we calculate the distance d_r as the opposite of spatial density and the hpt_r and tph_r (see Table 1) as occurrence frequency of head h and tail t, in which the d_r is the average mutual distance among the corresponding cluster of golden offsets. Then, we scatter each knowledge category in Fig. 5, and discover that if ch_r and ct_r for a specific relation r are almost the same (the marked area), the d_r is almost relatively small, while if the ch_r and ct_r differ greatly, the d_r is relatively large. In other words, the former entity-pair offsets $\hat{\boldsymbol{r}}$ cluster compactly and have high spatial density, while the latter entity-pair offsets $\hat{\boldsymbol{r}}$ cluster discretely and have low spatial density. Generalizedly, because of the diversity of occurrence frequency, i.e. imbalance and heterogeneity existing in KG, the cluster of $\hat{\boldsymbol{r}}$ for different categories will distribute with different density after the projection of TransE-like models.

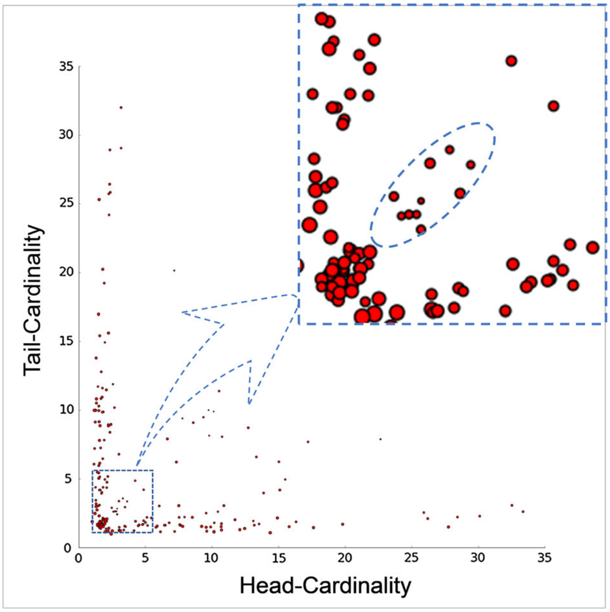

Fig. 5 The correlation between frequency of knowledge and spatial density. Each circle indicates a knowledge category. Its size depends on the average mutual distance d_r among the corresponding cluster of golden offsets $\boldsymbol{r}$, and the coordinate of each circle is dependent on the cardinality of head and tail arguments: ch_r and ct_r. Note that there are only 200 categories scattered in the figure over 2291 categories totally, but these categories contain 425464 triples in the total 483142 triples

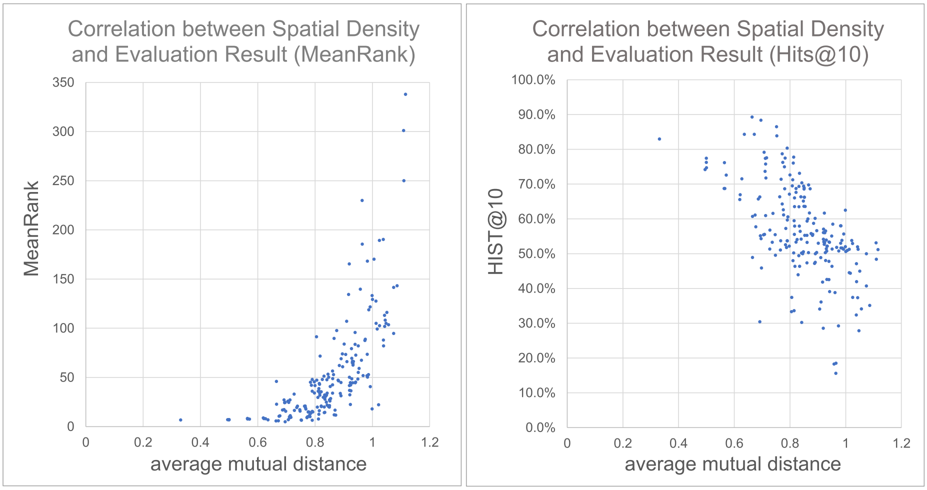

Fig. 6 The correlation between spatial density and evaluation result. Each dot indicates a category of knowledge and there are still 200 categories scattered in the figure

How spatial density affects embedding performance? To further explore the connection between the **category-specific density** and the embedding properties of distance-based models, we display the correlation between the spatial density and the evaluation result as shown in Fig. 6. The evaluation result is from the task of link prediction (Bordes et al. 2013) and there are two metrics: MeanRank and Hits@10, which will be elaborated in Section 5. The lower MeanRank or the higher Hist@10 gets, the better the KRL model performs.

Surprisingly, we discover that most of those categories with large d_r perform poor in the evaluation result, while those with small d_r perform well. The poor result is possibly caused by the inflexible separating margin and the inflexible importance trade-off between the golden Triple $\langle h, r, t\rangle$ and the synthetic Triple $\langle h', r, t\rangle$ or $\langle h, r, t'\rangle$, which are unsuitable for the category-specific density. For those knowledge categories with large d_r, the cluster of golden offsets $\hat{\boldsymbol{r}}$ distributes so discretely that the synthetic offsets $\hat{\boldsymbol{r}}'$ need to be further away from the golden cluster. Thus, the original fixed separating margin is too small to separate the synthetics $\hat{\boldsymbol{r}}'$ from the golden cluster, and the synthetic triple score function $f(h', r', t')$ need to be restricted more sufficiently.

Now that the spatial density for different knowledge categories differs a lot between each other, it should also be in diversity for the distance of margin to separate the goldens $f(h, r, t)$ and the synthetics $f(h', r', t')$ more appropriately. Motivated by the work of Wang et al. (2017) using an adaptive margin-based hinge loss function, we also adopt the margin adaptation and make the margin in our loss function adaptive to the spatial density of the representation. In this way, we can adaptively control the degree of separation between the goldens and the synthetics. The elaboration of this part is detailed in Section 4.

4 Our adaptive learning methods

In Section 3, we display the category-specific density and explain theoretically why we should adaptively choose the optimal margin and the optimal weight for each knowledge category to obtain better performance of embedding. In this section, we dive into the mathematical and algorithmic details of our adaptive learning methods and give a general framework that can be incorporated into any distance-based KRL model. Note that we take the clustering as the prerequisite of our proposed AWML framework.

4.1 Density-adaptive margin

When projecting the entities and relations into the embedding space, the distributed density of embedding points will differ from category to category, which derives from the imbalance and heterogeneity of KG. Consequently, an identical margin cannot be in the same order of magnitude with all the category-specific density, which will lead to a poor performance of embedding. It should also be in diversity for the distance of margin to separate the goldens $f(h, r, t)$ and the synthetics $f(h', r', t')$ more appropriately.

To overcome this shortcoming of previous KRL models, we also adopt a method of margin adaptation motivated by the work of Wang et al. (2017). In their work, an adaptive margin-based hinge loss function is used to improve the stability and performance of GANs. Similarly, an adaptive margin in the KRL training objective can also separate the goldens and the synthetics more appropriately to improve the embedding performance.

Furthermore, in that the spatial density is closely associated with the embedding performance (see Table 6) and is in the same spatial sense with the separating margin, we can make the separating margin adaptive to the spatial density of the representation. In this way, we can adaptively control the degree of separation between the goldens and the synthetics.

Therefore, in this work, we propose an **Adaptive Margin Learning (AML)** method, and the training objective is as follows and all loss-terms are divided by the number of summands in a batch:

$$L(S) = \sum_{\langle h,r,t\rangle \in S} \sum_{\langle h',r',t'\rangle \in S'_{\langle h,r,t\rangle}} [\gamma_r + f(h, r, t) - f(h', r', t')]_+, \tag{3}$$

The whole formation of training objective is the same as (1) except the separating margin γ_r that is adaptive to the category-specific density:

$$\gamma_r = \gamma_m \cdot \sigma(w_m \times dens_r^{-1} + b_m), \tag{4}$$

where the hyperparameter γ_m controls the whole range of the adaptive margin and put γ_r into the range from 0 to γ_m, $\sigma(\cdot)$ is a sigmoid function, and $w_m, b_m \in \mathbb{R}$ are the weight and bias parameters learned in the traning process.

The distributed density is inversely proportional to the average mutual distance of all the golden entity-pair offsets:

$$dens_r^{-1} = \frac{1}{|S_r|^2} \sum_{\langle h_1,r,t_1\rangle \in S_r} \sum_{\langle h_2,r,t_2\rangle \in S_r} \|\widetilde{\boldsymbol{r}}_{\langle h_1,t_1\rangle} - \widetilde{\boldsymbol{r}}_{\langle h_2,t_2\rangle}\|_{L_n}, \tag{5}$$

where S_r is the set of golden triples with the specific relation r, and $\widetilde{\boldsymbol{r}}_{\langle h,t\rangle}$ is the approximation of $\boldsymbol{r}$ with regard to $\boldsymbol{h}$ and $\boldsymbol{t}$, where the embedding vectors $\boldsymbol{h}$ and $\boldsymbol{t}$ are obtained with the pre-trained KRL model. Every KRL model has its distinctive approximation of $\boldsymbol{r}$ according to the distance-based score function. For instance, $\widetilde{\boldsymbol{r}}_{\langle h,t\rangle} = \boldsymbol{t} - \boldsymbol{h}$ in TransE, and $\widetilde{\boldsymbol{r}}_{\langle h,t\rangle} = \boldsymbol{t}\boldsymbol{M}_r - \boldsymbol{h}\boldsymbol{M}_r$ in TransR. Remark that the above calculation of density only suit for translation-based KRL models, and the calculation method for other distance-based KRL models will be discussed in Section 6.

In the training process of KRL model, when the average mutual distance is relatively large for some category, the separating margin will accordingly get larger to move the synthetic entity-pair offsets $\hat{\boldsymbol{r}}'$ further away from the relation embedding $\boldsymbol{r}$, and otherwise, the margin will get small.

4.2 Density-adaptive importance weight

In addition to adaptively controlling the separation between the goldens and the synthetics, we also consider the trade-off between the different contributions coming from the golden and synthetic triple among all the knowledge categories.

In that the under-restriction of either the golden triples or the synthetic triples exists in the previous KRL models, it is inflexible for the trade-off between the goldens' importance and the synthetics' importance, and their respective importance in the training objective should be finetuned. Based on this problem of inflexibility, we should control the contributions of these two restrictions: $f(h, r, t)$ and $f(h', r', t')$ flexibly.

Inspired by the work of Miyamoto and Cho (2016) and the work of Yang et al. (2016), we adopt an adaptive weight to control the contributions coming from the goldens and the synthetics in our proposed framework. In this way, the KRL training objective has ability to learn the goldens and the synthetics with adaptive importance.

Hence, in this work, the importance weights of golden-triple and synthetic-triple score function are introduced into the margin-based pairwise training objective. To adaptively select the optimal trade-off for every category and make it suitable for the category-specific density, we eventually propose another adaptive learning method, **Adaptive Weighted Learning (AWL)** to provide a framework to be incorporated into the KRL models. The training objective takes the form as:

$$L(S) = \sum_{\langle h,r,t\rangle \in S} \sum_{\langle h',r',t'\rangle \in S'_{\langle h,r,t\rangle}} [\gamma_u + (1 - \mu_r) f(h, r, t) - \mu_r f(h', r', t')]_+, \tag{6}$$

where the hyper-parameter γ_u are the weight and bias parameters. The two score function $f(h, r, t)$, $f(h', r', t')$ is mixed by a category-specific weight μ_r that depends on the density:

$$\mu_r = \frac{\beta + \sigma(w_u \times dens_r^{-1} + b_u)}{2\beta + 1}, \tag{7}$$

where $w_u, b_u \in \mathbb{R}$ is a bias scalar. The form of $\frac{\beta+\sigma(\cdot)}{2\beta+1}$ is to control the range of μ_r and let it around 0.5. The calculation of the density is the same as (5).

For some knowledge category with low density (i.e. large average mutual distance), the importance weight of synthetic triple score μ_r will get larger and the distance measurement of the synthetics will be more restricted, which lead to the result that the synthetic offset points will be further away from the golden cluster. If the weight μ_r get larger than the value of 0.5, the score function of synthetic triples will contribute more to the training objective than that of golden triples in the subsequent training process.

Table 5 Complexity of AWML framework

Model	#Parameters	#Time complexity
$\boldsymbol{\Psi_G}$	$O(N_p)$	$O(N_o)$
$\boldsymbol{\Psi_G}$ + AWL	$O(N_p + 2N_r)$	$O(\alpha N_o + \sum_{i=1}^{N_r} \binom{n_i}{2})$
$\boldsymbol{\Psi_G}$ + AML	$O(N_p + 2N_r)$	$O(\alpha N_o + \sum_{i=1}^{N_r} \binom{n_i}{2})$

N_r represents the number of relations. N_p and N_o represent the number of parameters and the time complexity of the baseline model Ψ_G, respectively

4.3 Training objective of AWML framework

In our AWML framework, the training objective is as follows:

$$\sum_{(\langle h',r',t'\rangle,\langle h',r',t'\rangle)\in T_{batch}} [\gamma_r + (1-\mu_r)f(h,r,t) + \mu_r f(h',r',t')]_+ \tag{8}$$

Among (8), γ_r denotes a density-adaptive margin if we choose the margin in adaption (AML framework), otherwise it denotes a fixed constant (a hyper-parameter). One category of knowledge has a specific margin, so it is relation-specific for the γ_r. It means that separating the goldens and the synthetics is adaptive to the knowledge category. Similarly, μ_r indicates a density-adaptive weight if we choose the AWL framework, otherwise it is a constant of 0.5.

As for the synthetic triples, they are constructed following (2), which differs from some other KRL models. In our synthetic-triple construction rule, the relation is considered additionally to corrupt the triple. It can make the KRL model appropriate also for triplet classification task,[5] not only for the link prediction.

The objective favors lower scores for golden triples as compared with synthetic triples, and it restricts the golden-triplet score function with the importance weight $1 - \mu_r$ and the synthetic-triplet one with the weight μ_r. If the category-specific weight μ_r gets larger than 0.5, then the model will try hard to maximize the synthetic triple score function, and the minimization of the golden triple score function will be pretty much ignored.

Algorithm implementation Algorithm 1 summarizes the whole AWML training process, and the margin or the weight can be chosen to be adaptive respectively.[6] The AWML framework initializes the entity and relation embeddings randomly (Bordes et al. 2013) or pretrainedly. We take the symbol of $\boldsymbol{\Psi_G}$ to denote any explicit KRL model, such as TransE. The learning method is decided by the $\boldsymbol{\Psi_G}$. For instance, in the work of TransE, the widely-used stochastic gradient descent (SGD) method is used to learn the embeddings, while in the work of Minervini et al. (2016), an adaptive learning approach, AdaGrad (Duchi et al. 2011), is utilized.

[5] We discover in our reproducing experiment that the original construction rule will make the KRL model perform poor in the classification task.

[6] Source code and datasets for reproducing the experiments presented in this paper are available online: https://github.com/orangegcc/AWML/

Algorithm 1 Learning AWML

```
Input: Some explicit KRL model $\boldsymbol{\Psi_G}$ over the knowledge graph $G$, training set $S = \{\langle h, r, t\rangle\}$, entity set $E$, relation set $R$, model hyper-parameters $\gamma_m$, $\gamma_u$, $\beta$, embedding dimension $k$, the boolean value p whether to use adaptive margin while (1-p) indicates whether to use adaptive weight.
Initailize
Step 1. Pre-train $\boldsymbol{\Psi_G}$ and cluster all the entity-pair offsets for each knowledge category to construct clustered relation set $R_c$.
Step 2. Initialize $\boldsymbol{\Psi_G}$: Initialize embeddings of each entity $e \in E$ and relation $r \in R_c$ randomly or with the embedding results of pre-trained $\boldsymbol{\Psi_G}$, normalize embeddings $\boldsymbol{r} \leftarrow \boldsymbol{r}/\|\boldsymbol{r}\|$ for each $r \in R_c$; Initialize the model parameters $w_m$, $b_m$ or $w_u$, $b_u$ randomly.
Step 3. Calculate each category-specific density according to (5).

loop
    $\boldsymbol{e} \leftarrow \boldsymbol{e}/\|\boldsymbol{e}\|$ for each $e \in E$
    $S_{batch} \leftarrow$ sample $(S, n)$ //sample a minibatch of size n
    $T_{batch} \leftarrow \emptyset$ //initialize the set of triple pairs

    $\mu_r \leftarrow 0.5$ for each $r \in R_c$
    Update density-adaptive margin $\gamma_r$ or density-adaptive weight $\mu_r$ for each $r \in R_c$ according to (4) or (7).

    for all $\langle h, r, t\rangle \in S_{batch}$ do
        $\langle h', r', t'\rangle \leftarrow sample(S'_{\langle h,r,t\rangle})$ //sample a corrupted triplet
        $T_{batch} \leftarrow T_{batch} \bigcup \langle h, r, t\rangle, \langle h', r', t'\rangle$
    end for
    Update embeddings and model parameters $w_m$, $b_m$ or $w_u$, $b_u$
    w.r.t. $\sum_{(\langle h',r',t'\rangle, \langle h',r',t'\rangle) \in T_{batch}} [\gamma_r + (1 - \mu_r) f(h, r, t) + \mu_r f(h', r', t')]_+$,
    where $f$ is the triplet score funtion of $\boldsymbol{\Psi_G}$
end loop

Output: All the entity embeddings $\boldsymbol{e}$ and relation embeddings $\boldsymbol{r}$
```

Before we begin to train the model, there are two things that should be conducted. Firstly, we should do the multi-projection on each relation. Particularly, we cluster all the entity-pair offsets $\boldsymbol{t} - \boldsymbol{h}$ for each knowledge category to construct clustered relation set R_c. In this way, one original relation has one or more than one sub-relations. Therefore, incorporated by our AWML framework, the number of relation embeddings will increase compared with the original KG (see Table 6). The second operation we should conduct is to calculate each category-specific density according to (5).

Afterwards, we can loop the training process following our training objective (see (8)). Among the objective, the triplet score function f is also decided by the specific KRL model, such as $f(h, r, t) = \|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|_{L_n}$ in TransE. In each epoch of training, we first normalize the entity embeddings.[7] Then we initialize the set of triple pairs – the goldens and the

[7] We only normalize the relation embedding in the first epoch. This is the same as the work of Bordes et al. (2013).

Table 6 Statistics of dataset and the results of clustering

Dataset	#Train	#Valid	#Test	#Ent	#Rel	#Rel after clustering		
						TransE	TransE(AG)	TransR
WN18	141442	5000	5000	40943	18	48	149	75
FB15k	483142	50000	59071	14951	1345	2291	2430	2467

synthetics following the synthetic-triple construction rule (see (2)). Last but not least, we calculate the loss based on the training objective (8) and update all the relation embeddings $\boldsymbol{r}$ and the entity embeddings $\boldsymbol{e}$. Additionally, for AML framework, the w_m and the b_m in (4) should be also updated so that the adaptive margin γ_r can change adaptively according to the precalculated density. As for AWL framework, w_u and the b_u in (7) should be also updated.

Comparisons of complexity Table 5 lists the complexities of the original KRL model $\boldsymbol{\Psi_G}$ and $\boldsymbol{\Psi_G}$+AWL/AML model. Compared with the baseline model CTransE, if it is incorporated by our adaptive framework, the number of parameters will be added by $\mathcal{O}(2N_r)$ because of the weight or the margin adaptive to the relation r. In that there are weight and bias parameters to be learned, N_r is multiplied by 2. The time complexity is similar between the $\boldsymbol{\Psi_G}$ and the $\boldsymbol{\Psi_G}$+AWL/AML except for 1) time concuming on weight or margin learning (so there is a factor $\alpha > 1$ before N_o) and 2) time consuming on density calculation $\mathcal{O}(\sum_{i=1}^{N_r}\binom{n_i}{2})$ before training, where the n_i denotes the number of triplets containing the same sub-relation. Hence, incorporated by our framework, the KRL model is still effective in time complexity.

5 Experiments

To evaluate our proposed framework AWML, we respectively incorporate AML and AWL into **3** cluster-based KRL models: CTransE, CTransE(AG) and CTransR, each of which is taken as the baseline in this work. Among the three cluster-based KRL models, their original approaches[8] before clustering are available respectively from TransE (Bordes et al. 2013), TransE(AdaGrad) (Minervini et al. 2016), which uses adaptive learning rate during representation learning, and TransR (Lin et al. 2015b), which adopts relational projection on entities. In the comparison experiments, we test the performances of **CTrans{E, E(AG), R} + AML** and **CTrans{E, E(AG), R} + AWL** for link prediction and triplet classification, and conduct visualization analysis on the embeddings.

Please note here that, it is CTransX model, not TransX model, that we compare CTransX+AWL/AML model with. So in the following table, we put the evaluating result of TransX in the brackets.

[8] Note that our evaluation results of TransE, TransE(AdaGrad), TransR and CTransR, may be different from the original works. This is because the synthetic-triple replacement rule in the loss function (see (2)) differs a lot from each other. In our framework, the relation is considered additionally in the rule to make the KRL model appropriate also for triplet classification task not merely for the link prediction. What's more, there exist some differences in the hyper-parameter settings between our framework and other works. We choose the best configuration of hyper-parameters in our experiments.

5.1 Experimental settings

Dataset The datasets we adopt are publicly available from two widely used knowledge graphs, WordNet (Miller 1995) and Freebase (Bollacker et al. 2008). WordNet is a lexical ontology for English language. In WordNet, each entity represents a synset consisting of several words, and a word can also belong to different synsets. Relationships between synsets include *hypernym*, *hyponym*, *meronym*, *holonym*, *troponym* and other lexical relations. As for Freebase, this large collaborative KG consists of a huge number of real-life facts and contains various entities such as people, places, events and so on. The WordNet and Freebase are so typical and popular that hundreds of Knowledge Representation Learning (KRL) models adopting this KG to evaluate the performance of models. Among all the subsets of WordNet and Freebase, we employ WN18 and FB15k used in Bordes et al. (2013) respectively, and their statistics are listed in Table 6.

In Section 3, we propose the relation multi-projection to disambiguate the relation with the method of clustering. After multi-projection, we can obtain more relational embeddings than that obtained from the original model $\boldsymbol{\Psi_G}$. Here we list the number of relations #Rel after clustering in Table 6.

Implementation details We train each evaluation model until it converges by using SGD(in mini-batch mode) for CTransE/CTransR and AdaGrad (Duchi et al. 2011) for CTransE(AG) with learning rate $\lambda = 0.01$. As for parameter regularization, we adopt the L_2 regularizer to all parameters for CTransE and CTransR on FB15k dataset, and for other evaluations we adopt the L_1 regularizer. Training time was limited to at most 2000 epochs over the training set. For three baseline models CTrans{E, E(AG), R}, we attempt several settings (Bordes et al. 2013; Minervini et al. 2016; Lin et al. 2015b) on the validation dataset to get the best configurations that are: dimension of embeddings $k = 20$, distance measurement $d = L_1$, the fixed margin $\gamma = 2$ for CTransE and CTransE(AG) on WN18; $k = 50$, $d = L_1, \gamma = 2$ for CTransR on WN18; $k = 50, d = L_2, \gamma = 0.5$ for CTransE and CTransR on FB15k; $k = 50, d = L_1, \gamma = 1.0$ for CTransE(AG) on FB15k.

For models incorporated by AWML framework, **CTransX + AML** and **CTransX + AWL**, we fix k and d that are the same as the settings of CTransE. Other hyper-parameters in our framework are: γ_m in AML, γ_u, β in AWL, and learning rate λ. We select λ from $\{0.01, 0.02\}$, γ_m from $\{1, 2, 4\}$, γ_u from $\{0.05, 0.25, 0.5\}$, β from $\{12, 24.5, 49.5\}$ to let μ_r and $1 - \mu_r$ not in a great disparity. We use the metric of MeanRank that is described in the following *Evaluation protocal* to select parameters on the validation set for both frameworks: AML and AWL, and for both initialization methods: randomly and pretrainedly. For CTransX + AWL the selected parameters are: $\lambda = 0.01$, $\gamma_u = 0.25$, $\beta = 24.5$ either randomly or pretrainedly initialized. Note that $\beta = 24.5$ will make the adaptive weight μ_r range from 4.9 to 5.1. For CTransX + AML the selected parameters are: $\lambda = 0.02$ when randomly initialized and $\lambda = 0.01$ when pretrainedly initialized, $\gamma_m = 2$ on WN18 dataset and $\gamma_m = 1$ on FB15k dataset.

5.2 Link prediction

Link prediction is a classical evaluation task that concentrates on the quality of knowledge representation (Bordes et al. 2013). This task aims to complete a triple when one of head or tail is missing, which can be viewed as a simple question answering task. Similar to the setting in Minervini et al. (2016) and Bordes et al. (2013), etc, the task returns a list of candidate entities from KG instead of one best answer.

Evaluation protocol We use two evaluating metrics by following Bordes et al. (2009): **MeanRank** and **Hist@n**. For each test triplet, we corrupt the head or tail by using other entities in the entity set E in turn and calculate the f scores for the test triplet and all the corrupted triplets. After that we rank these triplets with their scores by **descending** order. Finally, we get the ranking of correct entity. If the ranking of the correct entity is smaller than or equal to n, Hit@n for the test triplet will be equal to 1, otherwise it will be 0. For all the triplets in the testing data, we repeat the same procedure and get the MeanRank and mean value HITS@n for each kind of n $\in \{1, 3, 10\}$.[9] We report the average scores on head prediction and tail prediction as final evaluation results. It is clear that a good predictor has lower MeanRank and higher Hist@n_c.

When constructing corrupted triplets, some of them may hold in training or validation set, which indicates that they are also real triples. So we filter out these triples before the ranking of candidate entities. This evaluation setting is denoted as *Filt.* and *Raw* otherwise.

In the specific evaluation, we adopt our **novel ranking approach** appropriate for cluster-based models including CTransX, CTransX + AWL and CTransX + AML in this work. Because of multi-projection in cluster-based models, we do not know which **sub-relation** embedding should be used when calculating the triplet score f. To solve this problem, we firstly classify every corrupted entity pairs $\langle h, t\rangle$ into the sub-relation clusters by means of calculating which sub-relation embedding is the nearest neighbor of the entity-pair offset. Then we can use the corresponding **sub-relation** embedding to accomplish the link prediction task. Finally, the candidate entities can be ranked as the order in the former method. In the following, we call the two cluster-based metrics as MeanRank_c and Hist@n_c.

For CTransR, in that our framework with pretraining almost performs better than randomly initializing, we only do the experiments with pretraining. But we believe this is enough to demonstrate the generalization of our framework on the CTransR model.

Experimental results The overall results[10] of our frameworks as well as the baseline are shown in Table 7. On the dataset of FB15k, all settings of adaptive training bring a pronounced improvement to the original KRL model, no matter which triplet score function f we adopt (TransE or TransR), and no matter which learning method we use (SGD for CTransE or AdaGrad for CTransE(AG)). For the CTransE model, among all the incorporated approaches, AML with pre-trained initialization achieves the best MeanRank_c both in Raw[11] and Filt and also achieves the best HITS@n_c in 1, 3 and 10 settings. With AML, MeanRank_c(Filt.) of CTransE decreases by 10.2 and Hist@10_c increases by 6.0%. For the learning method of AdaGrad in CTransE(AG), we find that it is our AML framework that helps the CTransE(AG) achieve the best. Furthermore, we can discover that the result of HITS@n_c is robust to the value of n, which indicates that the performance of our proposed framework is insensitive to the evaluating metrics.

[9] Different from the formal evaluation metric HITS@10, we add the other two HITS@n to investigate the sensitivity of the performance to the HITS size.

[10] Note that, the models we compare contain CTransX(the baseline) and CTransX+AWL/AML, regardless of the original model TransX, including TransE, TransE(AG) and TransR. So all the evaluating results of TransX are not marked with bold font. Besides, to differentiate numerical values, we keep three decimal places for MeanRank_c of WN18 in the evaluation of Triplet Classification.

[11] Please note here that, the results of Raw setting differ greatly from other papers, this is derived from our modified ranking approach mentioned in the *Evaluation protocol*. When obtaining the f score for the test triplet with each candidate entity, we use **more than one** sub-relations to calculate the neighborhood score and choose the best one as the final sub-relation. So the correct head/tail will rank higher than that in the former evaluate method.

Table 7 Link Prediction Results: Test performance of CTransX(the baseline) and CTransX+AWL/AML on the WordNet(WN18) and Freebase(FB15k) KGs

Knowledge graph		Freebase(FB15k)					WordNet(WN18)				
METRIC		MeanRank_c		HITS@n_c(%)			MeanRank_c		HITS@n_c(%)		
Eval.setting		*Raw*	*Filt.*	*1*	*3*	*10*	*Raw*	*Filt.*	*1*	*3*	*10*
(TransE (ours))		(198.6)	(95.0)	(21.7)	(36.7)	(53.3)	(340.8)	(329.0)	(38.8)	(63.5)	(79.6)
CTransE		54.6	52.3	22.0	37.1	54.9	290.7	288.9	**31.7**	**55.9**	**76.8**
CTransE+AWL	*random*	51.4	47.9	24.6	39.3	57.0	469.0	467.1	12.6	36.2	56.2
	pretrain	51.1	47.6	24.9	39.5	57.1	**209.6**	**207.7**	22.5	42.8	63.2
CTransE+AML	*random*	51.2	47.4	25.8	40.0	58.3	532.3	530.5	7.5	25.6	54.1
	pretrain	**45.8**	**42.1**	**26.2**	**41.7**	**60.9**	296.3	294.6	29.1	52.6	74.5
(TransE(AG) (ours))		(216.1)	(90.5)	(21.2)	(38.1)	(56.9)	(605.3)	(593.0)	(33.6)	(57.8)	(74.1)
CTransE(AG)		68.5	64.8	23.7	38.6	57.2	383.4	381.4	14.2	48.5	72.5
CTransE(AG)+AWL	*random*	54.3	50.8	27.0	42.5	60.9	260.8	258.8	19.5	42.3	64.0
	pretrain	**53.5**	**49.9**	**27.1**	**42.8**	**61.3**	**204.6**	**202.7**	19.8	43.0	64.8
CTransE(AG)+AML	*random*	64.8	61.2	24.6	40.6	59.6	400.2	398.3	15.5	44.4	70.1
	pretrain	61.8	58.3	25.6	41.5	60.6	329.3	327.4	**25.2**	**52.4**	**75.7**
(TransR (ours))		(264.6)	(124.6)	(17.8)	(31.9)	(49.4)	(592.5)	(580.9)	(7.8)	(29.0)	(48.6)
CTransR		67.3	64.1	33.1	45.2	66.0	787.5	690.4	**6.0**	**24.1**	**43.4**
CTransR+AWL	*pretrain*	**63.5**	**59.6**	**35.3**	**46.7**	**67.1**	791.2	704.7	5.7	22.8	42.6
CTransR+AML	*pretrain*	66.8	63.3	33.4	45.2	66.5	**699.6**	**613.0**	5.9	23.5	42.9

Table 8 Link prediction on FB15k with respect to different types of relations(%)

Tasks	Predicting head (Hist@10_c)				Predicting tail (HITS@10_c)			
Relation type	1-1	1-M.	M.-1	M.-M.	1-1	1-M.	M.-1	M.-M.
CTransE	64.3	84.4	15.4	53.4	65	18.3	86.9	58.6
+AWL(random)	66.9	86.7	16.7	55.5	67.3	19.6	89.2	60.8
+AWL(pre-trained)	69.3	86.9	16.9	55.6	67.4	20.4	89.2	60.7
+AML(random)	65.5	88.4	16.0	57.2	64.5	20.8	89.8	62.4
+AML(pre-trained)	**70.7**	**90.6**	**18.2**	**59.9**	**68.7**	**23.9**	**92.1**	**64.9**

As for the WN18 KG, our framework also maintains comparative performance. Although the performance of the model with randomly initializing is somewhat poor, in all the compared models, the best MeanRank_c is consistently the model incorporated with our framework AWL or AML. In CTransE+AWL with pretrained setting, the MeanRank_c of Raw and Filt. settings are both decreases nearly 30%. Similarly, the CTransE(AG)+AWL also decreases a lot. This also indicates that our density-adaptive methods are consistently effective.

Additionally, as defined in Bordes et al. (2013), relations in KBs can be divided into four types according to hpt_r and tph_r (see Table 1): 1-to-1, 1-to-MANY, MANY-to-1 and MANY-to-MANY. Here we demonstrate the performance of AML and AWL incorporated into the baseline model on different types of relations in Table 8. We can observe that on all the **4** types of relations, both AML and AWL consistently achieve significant improvement as compared with the baseline, CTransE.

5.3 Triplet classification

We also test our model on triplet classification, which is used to evaluate the knowledge representation. This task aims to predict the missing relation given two entities and is equal to a classification task that classifies the testing triple into one of the knowledge categories. Similar to the evaluation protocol in link prediction, this task also returns a list of candidate relations from KG.

Evaluation protocol In this task, we corrupt the relation of each testing triple by using other relations in the set R in turn and calculate the f scores. After that we rank these triples in **descending** order. Similar to the link prediction task, we also use **MeanRank_c** and **Hist@n_c** for n ∈ {1, 3, 10} to evaluate the triplet classification results. Particularly, the HITS@1 metric indicates the classification accuracy and only check if the first relation in the sorted list is the correct one.

In the triplet classification task, we will not face the same problem that the multi-projection brings about. We only need to use all the sub-relation embeddings to replace the relation for each triplet. Besides, we use the Filt. setting in this task to compare all our frameworks with the baseline model.

Experimental results Evaluation results are shown in Table 9. We can see that on FB15k dataset, both AWL and AML models outperform CTransE on Hist@1 metric. On MeanRank_c metric, our AML model is slightly worse than CTransE, but our AWL

Table 9 Triplet Classification Results: Test performance of CTransX(the baseline) and CTransX+AWL/AML on the WordNet(WN18) and Freebase(FB15k) KGs

Knowledge graph		Freebase(FB15k)					WordNet(WN18)				
METRIC		MeanRank_c		HIST@n_c(%)			MeanRank_c		HIST@n_c(%)		
Eval.setting		*Raw*	*Filt.*	*1*	*3*	*10*	*Raw*	*Filt.*	*1*	*3*	*10*
(TransE (ours))		(3.05)	(2.75)	75.5	90.5	96.4	(1.219)	(1.215)	(90.9)	(97.6)	(99.9)
CTransE		3.32	3.05	75.5	92.9	97.0	1.227	1.224	91.6	**97.8**	99.5
CTransE+AWL	*random*	3.31	3.02	76.1	**93.1**	97.0	2.140	2.136	21.5	94.0	99.2
	pretrain	**3.26**	**2.97**	**76.3**	**93.1**	**97.2**	1.693	1.689	72.8	87.2	99.5
CTransE+AML	*random*	4.59	4.28	75.7	91.0	96.7	1.324	1.320	91.7	96.6	99.2
	pretrain	3.91	3.60	76.2	92.4	96.9	**1.216**	**1.212**	**92.5**	**97.8**	**99.7**
(TransE(AG) (ours))		(3.54)	(3.18)	(80.5)	(91.8)	(97.0)	(1.249)	(1.245)	(92.5)	(97.8)	(99.7)
CTransE(AG)		4.54	4.25	73.8	90.0	96.2	1.297	1.294	88.3	97.0	**99.6**
CTransE(AG)+AWL	*random*	3.37	3.08	**77.4**	91.3	97.2	1.571	1.567	83.2	91.6	**99.6**
	pretrain	**3.20**	**2.91**	77.3	91.3	97.3	1.600	1.597	82.1	91.3	**99.6**
CTransE(AG)+AML	*random*	4.09	3.77	75.3	91.4	96.9	**1.226**	**1.223**	**92.4**	97.9	99.5
	pretrain	3.52	3.20	75.7	**92.0**	**97.4**	1.236	1.232	91.8	**98.1**	99.5
(TransR (ours))		(5.24)	(4.86)	(81.6)	(93.1)	(97.2)	(1.232)	(1.228)	(94.0)	(97.3)	(99.4)
CTransR		4.73	4.41	75.3	92.2	97.6	1.893	1.889	92.2	96.9	**98.0**
CTransR+AWL	*pretrain*	**4.26**	**3.93**	**77.0**	**93.1**	**97.8**	2.031	1.993	91.5	96.4	97.6
CTransR+AML	*pretrain*	4.57	4.26	76.5	92.9	**97.8**	**1.762**	**1.738**	**92.3**	**97.1**	**98.0**

Table 10 Time consumption in the KRL training on WN18 and FB15k datasets

Model	Training time of 100 epochs (second)	
	WN18	FB15k
CTransE(AG)	1633.9	6640.7
CTransE(AG)+AWL	1916.4	8281.2
CTransE(AG)+AML	2050.0	7685.7

model achieves the best MeanRank_c 2.97 and the best Hist@1_c 76.3%. The improved Hist@1_c and the comparative MeanRank indicate that there are more testing triples classified into the correct knowledge category even though the relation of several triples are predicted extremely poorly. What's more, for any KRL model, our framework has the capability to help the KRL model achieve both the best MeanRank_c and the best Hist@n_c.

From the experimental results over both two tasks,[12] we can conclude that our AWML framework is capable of helping the KRL model to learn better entity and relation embeddings to accomplish the link prediction and the triplet classification.

5.4 Time efficiency analysis

In addition to the performance on two tasks, we also analyze the performance of our framework along the time efficiency. We list the time cost in the process of training for CTransE(AG) and CTransE(AG)+AWL/AML both on the datasets of WN18 and FB15k. As the time complexity in Section 4 analyzes, We can discover from Table 10 that the KRL model incorporated by our framework is similarly effective with the original model. The former is slightly more time consuming than the latter because of a factor $\alpha > 1$ shown in Table 5.

Note that, what we compare in Table 10 is the training time of the first 100 epochs not the training time of convergence epochs. Besides, the models of CTransE(AG)+AWL and CTransE(AG)+AL listed in Table 10 are all initialized randomly.

5.5 Visualization analysis

Our AWML framework makes the KRL model adaptive to the knowledge category to learn the embeddings. After we learn the embedding, we compare the representation distributions of CTransE and CTransE+AML through visualization. Similarly, we use t-SNE method (Maaten and Hinton 2008) to reduce the representations to 2-dim space. Then, we visualize all the Triples $\langle h, r, t \rangle$ for each knowledge category S_r and randomly pick 3 categories to

[12] We can discover from our evaluating results that "for CTransE model, AML is better than AWL for link prediction and AWL is better than AML for classification, but for some other models, it is contrary." This is because Link Prediction tends to be performed well by those embeddings that satisfy the condition that the head $\boldsymbol{h}$ is close to the vector of $\boldsymbol{t} - \boldsymbol{r}$, but Triplet Classification tends to be performed well by those embeddings that satisfy the condition that the relation $\boldsymbol{r}$ is being close to the vector of $\boldsymbol{t} - \boldsymbol{h}$. The thing worth mentioning is that **in the sense of average**, the above two conditions are not the sufficient and necessary between each other. Therefore, the performances on these two tasks are not absolutely the same. So for some KRL models, it can perform comparatively in one task but perform not so wonderfully in another.

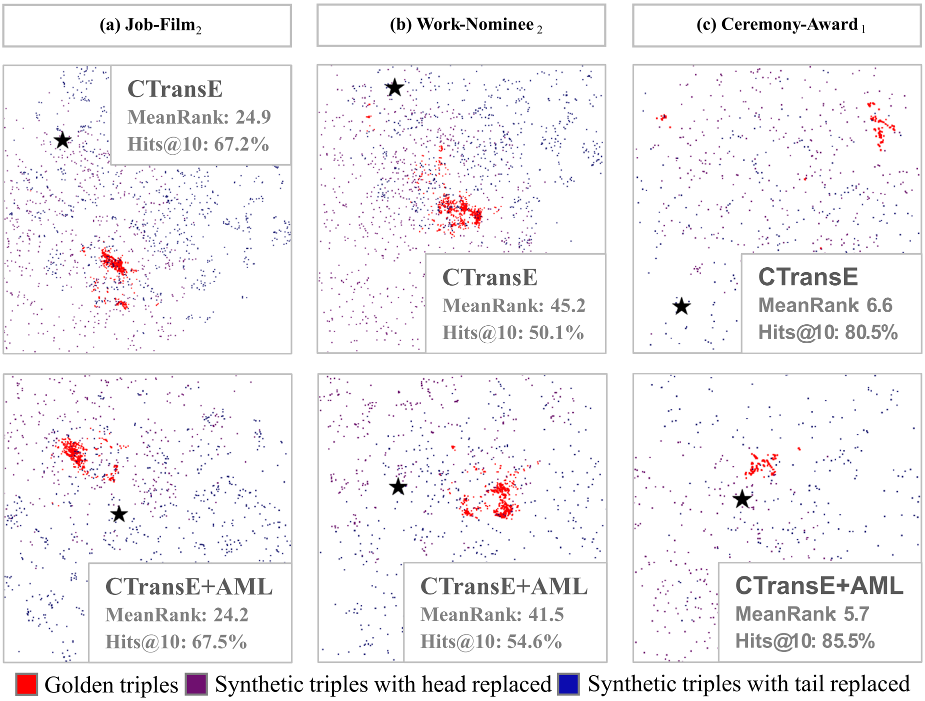

Fig. 7 Visualization results of CTransE embedding vectors with and without AML framework. Three relations ($a \sim c$) are randomly chosen from clustered relation set R_c that contains 2291 relations, and each clustered relation is denoted in the form of $Relation_N$. For each relation, two graphs are displayed to compare the CTransE+AML model and its baseline CTransE, and their link prediction results: MeanRank and HITS@10 are listed in the graph respectively. Each graph is visualized in the same size in the 2D plane. A black star denotes each relation embedding $\boldsymbol{r}$ and a colorful dot denotes the entity-pair offset $\boldsymbol{t} - \boldsymbol{h}$ of each triple as shown in the legend: red dot represents golden triple and dark dot represents synthetic triple

display in Fig. 7. Not only the goldens but also the synthetics we consider, and their positions are dependent on their entity-pair offset: the $\hat{\boldsymbol{r}}$ and the $\hat{\boldsymbol{r}}'$ respectively.

As shown in Fig. 7, for each category, the golden entity-pair offsets $\hat{\boldsymbol{r}}$ distribute similarly over CTransE+AML and CTransE.[13] But for CTransE+AML, the relation embedding $\boldsymbol{r}$ is much closer to the golden cluster, and at the same time, the evaluation result is also better than CTransE. This indicates that the embedding is obtained more appropriately to match the triple restriction of TransE: $\boldsymbol{t} - \boldsymbol{h} \approx \boldsymbol{r}$. In other words, in the distribution space, our adaptive framework is capable of building better representations to improve the performance of original KRL models.

[13] In Fig. 7, some synthetic triplets indeed spread around the relation embedding $\boldsymbol{r}$ after the CTransE model incorporated with our AWL framework. However, this phenomenon does not violate our expectation of the representation distribution, because it is in a sense of average and in a relative sense that the implicit vector of synthetic triplet should be further away from the relation embedding vector compared with the implicit vector of golden triplet. In the process of KRL training, in order to guarantee the total loss is low enough, the model tends to make a little of synthetic triples contrary to the above statement.

6 Discussion

In this section, we first analyze the necessity of multi-projection for KRL models. Then we provide the visualization analysis methods and discuss the approximation of $\boldsymbol{r}$ in (5) for other distance-based KRL models except for TransE-like models.

Multi-projection of relation and entity In Section 3.1, we propose the relation multi-projection to disambiguate the relation with the help of clustering. As a matter of fact, some other KRL models also imply the idea of multi-projection for entities, such as SE (Bordes et al. 2009), SLM (Socher et al. 2013), SME (Bordes et al. 2014), LFM (Jenatton et al. 2012; Sutskever 2009), NTN (Socher et al. 2013), RESCAL (Nickel et al. 2011; Nickel and Ring 2012), TransH (Wang et al. 2014), TransR/CTransR (Lin et al. 2015b), TransD (Ji et al. 2015). They all project head and tail entities into a relation-specific space when calculating the triple score function.

Why is the above entity multi-projection effective in building embeddings and improving the performance of KRL models? This is because the entity in KG also has the semantic ambiguity and the multi-projection will disambiguate the entity. Take SME as an example, for every triple $\langle h, r, t\rangle$, the model utilizes an relation-specific matrix $\boldsymbol{W_r}$ to transform the head embedding $\boldsymbol{h}$ into a relation-specific head embedding $\boldsymbol{h_r}$, and thus, when measuring different semantics associated with relations, an entity will be projected into distinctive embedding spaces to represent different contextual situations.

Therefore, no matter which concrete method we use, it is of great significance for KRL models to conduct the multi-projection.

Visualization analysis methods When we explore the training objective to analyze the performance of TransE in Section 3, we visualize the representation distributions using t-SNE. As for other KRL models, we can also utilize the similar visualization analysis methods to explore the distribution characteristics of embeddings. Here we provide concrete analysis methods for TransE-like models and other distance-based KRL models.

For TransE-like models that do not use the relation-specific matrix to multi-project the entity, such as TransA (Xiao et al. 2015), TransG (Xiao et al. 2016) and KG2E (He et al. 2015), we can visualize the embeddings with the same method as TransE — conduct the dimensionality reduction over all the relation and entity embeddings, take the same knowledge category as an observed collection, and analyze whether the entity-pair offset $\boldsymbol{t} - \boldsymbol{h}$ is in the neighborhood of the relation $\boldsymbol{r}$.

For TransE-like models with entity multi-projection, such as TransH (Wang et al. 2014), TransR/CTransR (Lin et al. 2015b), TransD (Ji et al. 2015), the entity embedding we visualize should be projected into the relation-specific space, otherwise the head, tail, and relation will not satisfy the translation property. Thus, we should conduct the multi-projection of entity embeddings before the dimensionality reduction so that we can then observe whether they satisfy the translation approximation $\boldsymbol{t}\boldsymbol{M}_{rt} - \boldsymbol{h}\boldsymbol{M}_{rh} \approx \boldsymbol{r}$.

For distance-based KRL models that project the nonlinear transformation of head and tail onto the relation embedding, such as NTN (Socher et al. 2013) and Hole (Nickel et al. 2015): $f(h, r, t) = g(\boldsymbol{r}^{\top} nl(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}))$, we can observe the distribution feature between the nonlinear transformation vector $nl(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$ and the relation embedding vector $\boldsymbol{r}$ for each triple. The projection vector of $nl(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$ on the $\boldsymbol{r}$ can be observed whether is small enough for the golden triples and large enough for the synthetics. For other models that utilize the bilinear transformation, such as LFM (Jenatton et al. 2012; Sutskever 2009) and RESCAL (Nickel et al. 2011; Nickel and Ring 2012), we can visualize the entity embeddings belonging to the same category of entity-pairs and analyze the distribution characteristics of the head and the tail in the bilinear transformation $\boldsymbol{h}^\top \boldsymbol{M}_r \boldsymbol{t}$.

Approximation of relation $\widetilde{r}_{\langle h,t \rangle}$ In Section 4.1, we display the approximation of $\boldsymbol{r}$ to calculate the distributed density $dens_r$ for TransE-like models. While for other distance-based KRL models, we can approximate the relation embedding $\boldsymbol{r}$ inspired by the score function f. For example in NTN model (Socher et al. 2013), the nonlinear transformation $tanh(\boldsymbol{h}, \boldsymbol{t})$ can be regarded as $\widetilde{\boldsymbol{r}}_{\langle h,t \rangle}$, because NTN consider the projection of $tanh(\boldsymbol{h}, \boldsymbol{t})$ on the $\boldsymbol{r}$ as the triple score function.

7 Conclusion and future work

In this paper, we tackle the knowledge embedding problem and propose an adaptive weighted margin learning framework, called AWML, to facilitate the KRL models to adaptively learn the representations of entities and relations in KG. We first analyze the visualization results of previous KRL model and discover the inconsistency between the original training objective and the complex property of KG. Then we explore the relation-specific density and explain the necessity of choosing an appropriate margin and importance weight for every knowledge category. Finally, we define the density-adaptive margin and the density-adaptive weight, and integrate them into the previous training objective respectively for knowledge embedding. Experimental and visualized results validate the effectiveness of our proposed framework.

From the performances on the evaluation tasks, we can conclude that our proposed framework is indeed capable of helping the KRL models to obtain better representations in the embedding space. The good performance is derived from the ability of adaptation. With our framework, the KRL model can adaptively control the contributions of golden and synthetic triples in the training process, and also can adaptively control the degree of separating the two kinds of restrictions. But there still exist challenges for our proposed KRL framework. For the 1-to-N and N-to-1 relations, it is still very difficult to learn a perfect representation and the improvement is small. For another, with our framework, some category of triples distribute worse in the embedding space. As the visualization shows in Fig. 7, we can see that there are still some synthetic implicit vectors distribute near the relation embedding. These two challenges are still what we should explore in the future work. Additionally, we will incorporate our framework, AWML, into more KRL models and apply it in more tasks to evaluate the generalization of our framework. It is possible to focus on the entity rather than the relation to analyze the distribution characteristics of the embeddings and explore the capability of knowledge embedding models.

Acknowledgements This research is supported by the National Natural Science Foundation of China under Grant No.61602048, No.61601046 and No.61520106007.

References

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD 08 Proceedings of the 2008 ACM SIGMOD international conference on management of data* (pp. 1247–1250).

Bordes, A., Glorot, X., Weston, J., Bengio, Y. (2012). Joint learning of words and meaning representations for open-text semantic parsing. *International Conference on Artificial Intelligence & Statistics*, *22*, 127–135.

Bordes, A., Glorot, X., Weston, J., Bengio, Y. (2014). A semantic matching energy function for learning with multi-relational data: application to word-sense disambiguation. *Machine Learning*, *94*(2), 233–259.

Bordes, A., Usunier, N., Weston, J., Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Advances in NIPS*, *26*, 2787–2795.

Bordes, A., Weston, J., Collobert, R., Bengio, Y. (2009). Learning structured embeddings of knowledge bases. *Aaai Conference on Artificial Intelligence*, (Bengio), 301–306.

Boser, B.E., Guyon, I.M., Vapnik, V.N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory - COLT '92* (pp. 144–152).

Duchi, J., Hazan, E., Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, *12*(1532-4435), 2121–2159.

Ferràndez, A., Matė, A., Peral, J., Trujillo, J., De Gregorio, E., Aufaure, M.A. (2016). A framework for enriching data warehouse analysis with question answering systems. *Journal of Intelligent Information Systems*, *46*(1), 61–82.

Han, X., Zhang, C., Guo, C. (2018). A generalization of recurrent neural networks for graph embedding. In *Proceedings of the 22nd Pacific-Asia conference on knowledge discovery and data mining*. Melbourne.

He, S., Liu, K., Ji, G., Zhao, J. (2015). Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM international on conference on information and knowledge management - CIKM '15* (pp. 623–632).

Jenatton, R., Bordes, A., Roux, N.L., Obozinski, G. (2012). A latent factor model for highly multi-relational data. *Advances in Neural Information Processing Systems*, *25*, 3167–3175.

Ji, G., He, S., Xu, L., Liu, K., Zhao, J. (2015). 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, pp. 687–696).

Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S. (2015a). Modeling relation paths for representation learning of knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 705–714). Stroudsburg: Association for Computational Linguistics.

Lin, Y., Liu, Z., Zhu, X., Zhu, X., Zhu, X. (2015b). Learning entity and relation embeddings for knowledge graph completion. In *Twenty-Ninth AAAI conference on artificial intelligence* (pp. 2181–2187).

Maaten, L.V.D., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research 1*, *620*(1), 267–284.

Metzger, S., Schenkel, R., Sydow, M. (2017). QBEES: query-by-example entity search in semantic knowledge graphs based on maximal aspects, diversity-awareness and relaxation. *Journal of Intelligent Information Systems*, *49*(3), 333–366.

Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *NIPS*, (pp. 1–9).

Miller, G.A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Minervini, P., D'Amato, C., Fanizzi, N. (2016). Efficient energy-based embedding models for link prediction in knowledge graphs. *Journal of Intelligent Information Systems*, *47*(1), 91–109.

Miyamoto, Y., & Cho, K. (2016). Gated word-character recurrent language model, 1992–1997.

Nickel, M., & Ring, O. (2012). Factorizing YAGO scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web* (pp. 271–280).

Nickel, M., Tresp, V., Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *ICML*, (pp. 809–816).

Nickel, M., Rosasco, L., Poggio, T. (2015). Holographic embeddings of knowledge graphs. In *Thirtieth Aaai conference on artificial intelligence*.

Shi, B., & Weninger, T. (2017). ProjE: embedding projection for knowledge graph completion. In *AAAI*.

Socher, R., Chen, D., Manning, C., Chen, D., Ng, A. (2013). Reasoning with neural tensor networks for knowledge base completion. In *Neural information processing systems 2003* (pp. 926-934).

Sutskever, I. (2009). Modelling relational data using Bayesian clustered tensor factorization. *Nips*, *22*, 1–8.

Wang, R., Cully, A., Chang, H.J., Demiris, Y. (2017). MAGAN: margin adaptation for generative adversarial networks.

Wang, Z., Zhang, J., Feng, J., Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *AAAI conference on artificial intelligence* (pp. 1112–1119).

Weston, J., & Watkins, C. (1999). Support vector machines for multi-class pattern recognition. In *Proceedings of the 7th European symposium on artificial neural networks (ESANN-99)* (pp. 219–224).

Xiao, H., Huang, M., Hao, Y., Zhu, X. (2015). TransA: an adaptive approach for knowledge graph embedding. arXiv:1509.0.

Xiao, H., Huang, M., Yu, H., Zhu, X. (2016). TransG: a generative mixture model for knowledge graph embedding. In *Proceedings of ACL* (pp. 2316–2325).

Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M. (2016). Representation learning of knowledge graphs with entity descriptions. *Aaai*, 2659–2665.

Yang, Z., Dhingra, B., Yuan, Y., Hu, J., Cohen, W.W., Salakhutdinov, R. (2016). Words or characters? Fine-grained gating for reading comprehension.

Zhang, C., Zhou, M., Han, X., Hu, Z., Ji, Y. (2017). Knowledge graph embedding for hyper-relational data. *Tsinghua Science and Technology*, *22*(2), 185–197.

Zhao, F., Min, M.R., Shen, C., Chakraborty, A. (2017). Convolutional neural knowledge graph learning. arXiv:1710.0.

Zhou, M., Zhang, C., Han, X., Ji, Y., Hu, Z., Qiu, X. (2016). Knowledge graph completion for hyper-relational data. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 9784, pp. 236–246).

Affiliations

Chenchen Guo[1] · Chunhong Zhang[1] · Xiao Han[1] · Yang Ji[1]

Chenchen Guo
orangegcc@bupt.edu.cn

Xiao Han
hanxiao1007@bupt.edu.cn

Yang Ji
jiyang@bupt.edu.cn

[1] Key Laboratory of Universal Wireless Communications, Ministry of Education,
Beijing University of Posts and Telecommunications, Beijing, China