# Attention-based Learning for Multiple Relation Patterns in Knowledge Graph Embedding

**Anonymous ACL submission**

## Abstract

Relations in knowledge graphs often exhibit multiple relation patterns. Various knowledge graph embedding methods have been proposed to modeling the properties in relation patterns. However, relations with a certain relation pattern actually only account for a small proportion in the knowledge graph. Relations with no explicit relation patterns also show complicated properties which is rarely studied. To this end, we argue that a property of a relation should either be *global* or be *partial*, and propose an **A**ttention-based **L**earning framework for **M**ulti-relation **P**atterns (ALMP) for expressing complex properties of relations. ALMP adopts a set of affine transformations for expressing corresponding global relation properties. Furthermore, ALMP utilizes a module of attention mechanism to integrate the representations. Experimental results show that the ALMP model outperforms baseline models on the link prediction task.

## 1 Introduction

Deep learning has made amazing progress in the past decade and is facing an important transition from an intuitive and perceptive black-box (system1) to a conscious and logical system (system2) (Bengio et al., 2021; Bengio, 2019). Meanwhile, knowledge graphs (KG), a data structure depicting the correlation of real word entities, is regarded as an essential part of system2, since it enhanced reasoning capability and interpretability by combining knowledge and intelligence. Therefore, learning the representation of knowledge graphs in vector space for downstream deep learning systems becomes a task that attracts much attention.

The most important feature that distinguishes knowledge graphs from general graphs is that edges between nodes (entities) in KGs represent multiple relations, which contain their unique properties. Recently, some KG embedding methods introduced a concept of *relation pattern* (e.g. symmetry, transitivity, etc.) to describe the consistent property which a relation exhibits on all instances in KG (Sun et al., 2019). Furthermore, for capturing these properties, existing methods try to model the relation as a certain mapping approach between the head and tail entities. For example, RotatE (Sun et al., 2019) defines each relation as a rotation from the head entity to the tail entity to modeling and inferring relation patterns like symmetry/asymmetry, inversion, and composition. Such modeling principle is consistent with the idea of *relational inductive biases* proposed by (Battaglia et al., 2018), which enables relational reasoning by imposing constraints on the relations as well as the interactions among entities.

The relation with a certain relation pattern actually means that all instances related to the relation satisfy the rule form of the relation pattern. Here we use *global relation property* to refer to such pattern of relations. However, most existing embedding methods actually pay less attention to relations with multiple global properties simultaneously.

On the other hand, in most KGs, the amount of relations with no explicit relation patterns is far larger than relations with a certain pattern. As Figure 1 shows, there are 91.9% of relations with undefined relation patterns in YAGO3 (Mahdisoltani and Suchanek, 2015) [1]. In fact, these relations may also follow some relation patterns on some/all observed facts but violate the patterns on other/unobserved facts. In other words, a relation may show several relation patterns simultaneously on its different instances. We use *partial relation property* to refer to such properties that hold over some subsets of entities but not all.

Therefore, a pertinent question for KG embedding methods is: can we learn an integrated repre-

[1]The statistic result is according to the YAGO3 schema: https://github.com/yago-naga/yago3/tree/master/schema.

sentation which combines various multiple relation properties so that it could better express the complicated interactions between entities?

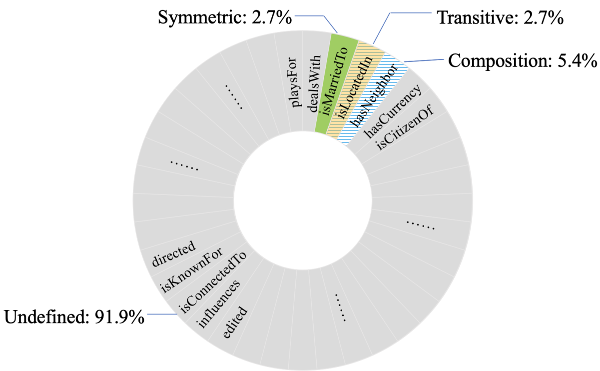



Figure 1: The proportion of relations with explicit relation pattern defined on YAGO3-10.

To this end, we argue the relation patterns that a relation has should be either be *global*, which means that *every* instances satisfy the relation pattern; or *partial*, which means that only *partial* instances satisfy the relation pattern. We further propose a novel framework based on KG embedding with affine transformations, namely the **A**ttention-based **L**earning for **M**ultiple relation **P**atterns (ALMP). The ALMP model is inspired by various relational inductive biases imposed by the KG embedding models according to different relation patterns. We systematically combine the geometric transformations prior with the properties of relation patterns from the perspective of relational inductive biases. Then, we learn integrated KG representation utilizing attention mechanism to incorporate features of various relation properties. Finally, we show experimental results on the link prediction task over three benchmarks, where ALMP has better performance comparing to the baseline methods with single relational inductive biases.

## 2 Related Work

We categorize KG embedding models into the following different types according to the approaches they choose to utilize relational inductive biases.

**Translation as relational inductive bias.** KG embedding models of this type implicitly impose an inductive bias as modeling the relation as a vector addition from a head entity to a tail entity. The well-known series of models in KG embedding area are the translation-based models represented by TransE (Bordes et al., 2013). TransE proposed a distance-based scoring function, which assumes the added embedding of subject entity $h$ and relation $r$ should be close to the embedding of object entity $t$. To solve the 1-To-N problem in TransE, variants of translational architectures have been developed. TransH (Wang et al., 2014) projects entities and relations into a relation-specific hyperplanes, which enables different projections of an entity in different relations. TransR (Yankai Lin and Zhu, 2015) introduces relation-specific spaces, which builds entity and relation embeddings in different spaces separately. A recent model BoxE (Abboud et al., 2020), embeds entities as points, and relations as a set of boxes, for yielding a model that could express multiple relation patterns.

**Linear mapping as relational inductive biases.** KG embedding models of this type modeling the relations as linear mappings from head entities to tail. DistMult (B. Yang and Deng, 2015) model the relation as a bilinear diagonal matrix between head and tail entities for multiple relational representation learning. To expand Euclidean space, ComplEx (T. Trouillon and Bouchard, 2016) firstly introduces complex vector space which can capture both symmetric and asymmetric relations. Similarly, RotatE (Sun et al., 2019) models in complex space and can capture additional inversion and composition patterns by introducing rotational Hadamard product. Extending the embedding from complex space to quaternary space, QuatE (Zhang et al., 2019a) use a quaternion inner product and gains more expressive semantic learning capability. Tucker (Balazevic et al., 2019) utilize Tucker decomposition of the binary tensor representation of triples. Recently, PairRE (Chao et al., 2021) proposed a method to model each relation with paired vectors to project the corresponding head and tail entities for better handle multiple relation patterns. To remedy the drawback that previous models cannot model the transitive relation pattern, Rot-Pro (Song et al., 2021) imposes projection on both source and target entities for expressing transitivity, and utilize a rotation operation as RotatE to underpin other relation patterns.

**Attention mechanism as relational inductive biases.** KBGAT (Nathani et al., 2019) is an attention-based embedding model that captures both entity and relation features of neighborhoods of any given entities. The latest model GAATs

| Relation pattern | Rule form | LT form | TT form |
|---|---|---|---|
| Symmetry | $r(x,y) \Rightarrow r(y,x)$ | $\mathbf{M}_r\mathbf{M}_r = \mathbf{I}$ | $\mathbf{r}+\mathbf{r}=0$ |
| Asymmetry | $r(x,y) \Rightarrow \neg r(y,x)$ | $\mathbf{M}_r\mathbf{M}_r \neq \mathbf{I}$ | $\mathbf{r}+\mathbf{r}\neq 0$ |
| Inversion | $r_2(x,y) \Rightarrow r_1(y,x)$ | $\mathbf{M}_{r_1}\mathbf{M}_{r_2} = \mathbf{I}$ | $\mathbf{r}_1+\mathbf{r}_2=0$ |
| Composition | $r_2(x,y) \wedge r_3(y,z) \Rightarrow r_1(x,z)$ | $\mathbf{M}_{r_2}\mathbf{M}_{r_3} = \mathbf{M}_{r_1}$ | $\mathbf{r}_1+\mathbf{r}_2=\mathbf{r}_3$ |
| Transitivity | $r(x,y) \wedge r(y,z) \Rightarrow r(x,z)$ | $\mathbf{M}_r^n = \mathbf{M}_r$ | $n\mathbf{r}=\mathbf{r}$ |

Table 1: Rule form of relation patterns and the attributes of their corresponding mapping matrix, where LT form represents linear transformation form (Zhang et al., 2019b) and TT represents translation transformation.

(Wang et al., 2020) integrates an attenuated attention mechanism to assign different weight in different relation path and acquire the information from the neighborhoods so that entities and relations can be learned in any neighbors. Beyond the scope of graph neural network, ATTH (Chami et al., 2020) recently proposes a low-dimensional hyperbolic knowledge graph embedding method to capture tree-like structures and hence modeling hierarchy data. It further conducts attention-based transformations of reflection and rotation for multiple relation patterns, which is similar to our proposal. The main difference between ATTH and our method is that ATTH focused on the hyperbolic embedding for hierarchical data, while we emphasize on integrating multiple transformations for modeling complex interactions among different relation patterns.

## 3 Multiple relation property problem

Relation patterns play important role in KG completion because missing/unobserved facts can be inferred based on these patterns. Existing methods dedicate a lot to model such patterns. A general methods is regarding relations as translation or linear transformations from head entity to tail entity. We list the five common relation patterns mentioned on previous work in Table 1, and the corresponding linear or translation transformation form that could model these patterns.

However, as mentioned above, most relations in KGs do not exhibit an explicit pattern, and hence these KG embedding methods may tend to over fit for a certain relation pattern since the model forces all relations to follow a certain transformation. As Battaglia et al. point out that, ideally, inductive biases both improve the search for solutions as well as finding solutions that generalize in a desirable way, however, when the introduced inductive biases are too strong, it tends to lead to sub-optimal performance (Battaglia et al., 2018).

Therefore, we seek to explore the multiple relation property problem. We observed that the multiple relation property problem can be divided into two circumstances:

(1) Multiple *global relation properties* (i.e. relation patterns) can exhibit in a relation simultaneously. For example, relation *isLocatedIn* in YAGO3-10 describes the relations of geographical locations. Obviously, it shows global transitive as well as asymmetric property among all its instances.

(2) Relations with no explicit relation pattern may also show one/multiple *partial relation properties* over some subsets of entities. For example, relation *isConnectedTo* in YAGO3 describes the connectivity between different airports. It exhibits partial symmetry or transitivity pattern on certain subsets airports.

Current models with so-called fully expressiveness mainly focus on modeling single relation pattern. For example, in Rot-Pro (Song et al., 2021), the solution space of modeling transitivity is that when the relational rotation phase is $2n\pi$ while that of modeling symmetric is that when the relational rotation phase is $n\pi$. Therefore, it theoretically could not modeling relations with both transitivity and asymmetric pattern like *isLocatedIn*.

An intuitive way to solve the multiple relation property problem is to construct a higher dimensional vector space, and define numerous hyperplane to map the entity representations into the space specific to these patterns, which is similar to (Wang et al., 2014; Yankai Lin and Zhu, 2015). However, this approach will introduce a large number of parameters which may cause decrease in time efficiency and does not consider the properties of multiple relation patterns occurred in a single relation. Therefore, we introduce a generic framework ALMP to integrate the multiple representations of various relation properties.

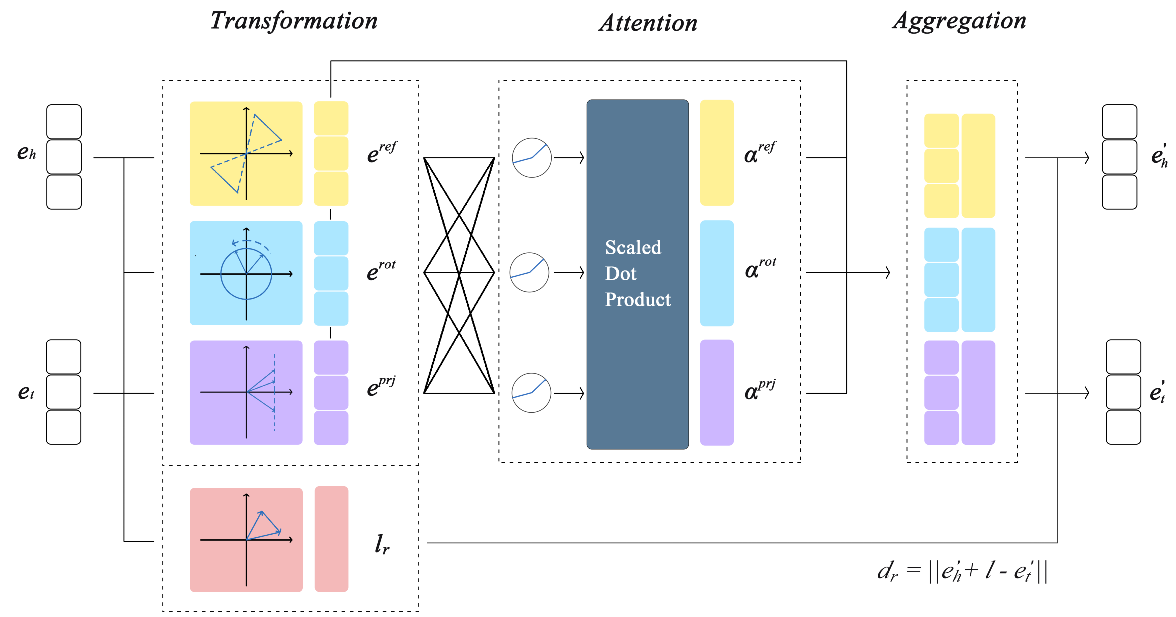



Figure 2: The structure of the ALMP framework. The transformation module obtains the entity embedding via three linear transformation as well as the translation. Then the attention module learns the attention weight assigned to each element of each transformed embedding. Finally, the aggregation module obtains the final embedding by integrating the attention with the corresponding embeddings.

## 4 Attention-based learning for multiple relation patterns

### 4.1 Parameterization

We parameterize the the embedding of entity and relation in 2D vector space and denote them by $\mathbf{e}$ and $\mathbf{e}_r$ respectively. The embedding dimension is an even number $d$, then the set of parameters $\Theta := \{\Theta_e, \Theta_r\}$.

$$\Theta_e := \begin{bmatrix} \mathbf{e}_i^x \\ \mathbf{e}_j^y \end{bmatrix}, (i \in \{1, ..., \frac{d}{2}\}, j = i + \frac{d}{2}), \quad (1)$$

where $e_i^x$ and $e_j^y$ are the corresponding components on each dimension of the x and y axis.

The relation parameterization is composed of the following components:

$$\Theta_r := \begin{cases} \Theta_{r,i}^{\text{ref}} = diag\left(T_{r,i}^{\text{ref}}(\theta_{r_1})\right), \\ \Theta_{r,i}^{\text{rot}} = diag\left(T_{r,i}^{\text{rot}}(\theta_{r_2})\right), \\ \Theta_{r,i}^{\text{prj}} = diag\left(T_{r,i}^{\text{prj}}(\theta_{r_3}, a_r, b_r)\right), \\ \Theta_{r,i}^{l} = l_{r,i}, (i \in \{1, \ldots, \frac{d}{2}\}) \end{cases} \quad (2)$$

where $\Theta_{r,i}^{\text{ref}}, \Theta_{r,i}^{\text{rot}}$ , and $\Theta_{r,i}^{\text{prj}}$ are the geometric-specific parameters on each dimension $i$, and $T_{r,i}^{\text{ref}}(\theta_{r_1}), T_{r,i}^{\text{rot}}(\theta_{r_2})$, and $T_{r,i}^{\text{prj}}(\theta_{r_3}, a_r, b_r)$ are the 2D matrix form of each geometric transformation. We will illustrate the geometric details in the next section. Meanwhile, in the rest of paper, we will omit the dimensional index $i$ in vectors for simplicity.

### 4.2 Core modules

The general structure of the ALMP framework is illustrated in Figure 2, which contains the following core modules.

**Transformation module.** We uniformly taking relations as four representative affine transformations, which are translation, reflection, rotation, for capturing various partial properties in relations. The reason for taking relations as affine transformations is that they could naturally express geometric operations and fit the different partial properties of relations. The four transformations and their corresponding properties that they could capture are illustrated in the following items:

- **Reflection**: An informal description of reflection in KG embedding is that: the head entity will return to itself after transforming twice. Therefore, it could naturally represent *sym-*

*metric* relation pattern geometrically [2]. According to the linear algebra theorem (Valenza, 2012), the corresponding 2D matrix form of reflection in Equation 2 is as follows:

$$T_r^{\text{ref}}(\theta_{r_1}) = \begin{bmatrix} \cos\theta_{r_1} & \sin\theta_{r_1} \\ -\sin\theta_{r_1} & \cos\theta_{r_1} \end{bmatrix} \quad (3)$$

- **Rotation**: Regarding relations as rotations from head entities to tail entities could naturally model *inverse*, *asymmetry* and *composition* patterns since the relation with such patterns involve the matching with other relations. RotatE (Sun et al., 2019) utilize the relational rotation in complex space, which is analogous with 2D euclidean space. The matrix form of rotation in Equation 2 is:

$$T_r^{\text{rot}}(\theta_{r_2}) = \begin{bmatrix} \cos\theta_{r_2} & -\sin\theta_{r_2} \\ \sin\theta_{r_2} & \cos\theta_{r_2} \end{bmatrix} \quad (4)$$

- **Projection**: Projection in vector space is equivalent to the idempotent transformation, which could express the *transitivity* relation pattern. However, conducting projection on a vector will cause the loss of dimensional information. Therefore, models such as Rot-Pro (Song et al., 2021) expressed projection in the form of similarity of idempotent transformation. According to Rot-Pro, the matrix form of projection in Equation 2 is defined as:

$$T_r^{\text{prj}}(\theta_{r_3}, a_r, b_r) = S_r^{-1}(\theta_{r_3}) \begin{bmatrix} a_r & 0 \\ 0 & b_r \end{bmatrix} S_r(\theta_{r_3}), \quad (5)$$

  where $S_r(\theta_{r_3})$ is an invertible matrix with parameter $\theta_{r_3}$, and $a_r, b_r \in \{0, 1\}$.

- **Translation**: The corresponding geometric operation of translation in vector space is the addition of vector ($l_r$ in Equation 2). Translation could model relation patterns such as *asymmetry*, *inversion*, and *composition*.

The initial embeddings of the head and tail entity $h, t$ are denoted by $\mathbf{e}_h, \mathbf{e}_t$, which are obtained via a shallow encoder[3]. Then $\mathbf{e}_h$ and $\mathbf{e}_t$ are simultaneously transformed by three types of linear transformations, which represents reflection, rotation, and projection respectively.

Theoretically, each transformation is prone to learn independently the corresponding relation patterns which fit itself. We use the form:

$$\mathbf{e}_h^{\text{ref}} = T_r^{\text{ref}}(\mathbf{e}_h), \mathbf{e}_h^{\text{rot}} = T_r^{\text{rot}}(\mathbf{e}_h), \mathbf{e}_h^{\text{prj}} = T_r^{\text{prj}}(\mathbf{e}_h)$$

$$\mathbf{e}_t^{\text{ref}} = \mathbf{e}_t, \mathbf{e}_t^{\text{rot}} = \mathbf{e}_t, \mathbf{e}_t^{\text{prj}} = T_r^{\text{prj}}(\mathbf{e}_t)$$

to denote the transformed embeddings of $\mathbf{e}_h$ and $\mathbf{e}_t$ after reflection, rotation and projection. Note that due to the principle of transitivity modeling, the projection operation should be conducted simultaneously on both head and tail entities.

**Attention module.** For integrating the expressiveness of the three embeddings aforementioned, it is natural to utilize attention mechanism to focus on specific transformation that fits the relation pattern (Chami et al., 2020). Here we employ an element-wise attention, which learns the attention weights on each dimension, since we assume that each dimension of a well-learned representation is a disentangled factor and should be assigned with different attention weight from different relation patterns. The attention weight can be obtained based on the following equation.

$$[\alpha^{\text{ref}}; \alpha^{\text{rot}}; \alpha^{\text{prj}}] = \sigma(\mathbf{W}_r \cdot [\mathbf{e}^{\text{ref}}; \mathbf{e}^{\text{rot}}; \mathbf{e}^{\text{prj}}]), \quad (6)$$

where $\mathbf{W}_r \in \mathbb{R}^d$ is a trainable vector; $[\cdot;\cdot]$ denotes the concatenation operation; The vector $\alpha^{\text{ref}}$, $\alpha^{\text{rot}}$, and $\alpha^{\text{prj}} \in \mathbb{R}^d$, and each $\alpha_i^{\nu}$ scores how much the $i$-th component of the embedding is related to the corresponding transformation ($\nu \in \{\text{ref}, \text{rot}, \text{prj}\}$); and $\sigma$ refers to an non-linear activation function such as softmax.

**Aggregation module.** Based on various linear transformation and the attention mechanism, we have obtained the three transformed embeddings along with their corresponding element-wise attention weights. To integrate them together, multiple aggregating methods could be considered as long as it is permutation invariant (e.g. summation or average over $\{\text{ref}, \text{rot}, \text{prj}\}$). The general form of the aggregation can be defined as follows:

$$\mathbf{e}' = agg(\alpha^{\nu} \odot \mathbf{e}^{\nu}), \quad (7)$$

where $\nu \in \{\text{ref}, \text{rot}, \text{prj}\}$ and $\odot$ denotes the Hadamard product.

[2]Note that relational rotation can model symmetric pattern only when the relational rotation phase is $n\pi$, ($n = 0, 1, 2, \ldots$). While reflection is more general and straightforward for modeling symmetric pattern.

[3]A shallow encoder in KG embedding can be viewed as a lookup function that finds the hidden representation corresponding to an entity or a relation given its index (Kazemi et al., 2020).

### 4.3 Scoring function

For each triple $(h, r, t)$, the distance function of the ALMP model is defined as the following form:

$$d_r(\mathbf{e}_h, \mathbf{e}_t) = \|agg(\alpha^\nu \odot \mathbf{e}_h^\nu) + l_r - agg(\alpha^\nu \odot \mathbf{e}_t^\nu)\|,$$

where $l_r$ is a vector in $\mathbb{R}^d$ to integrate translation transformation for relation $r$. The final scoring function is defined as:

$$f_r(\mathbf{e}_h, \mathbf{e}_t) = -d_r(\mathbf{e}_h, \mathbf{e}_t) + b_h + b_t, \quad (8)$$

where $b_h$ and $b_t$ are the head and tail entity biases that act as margins in the scoring function (Chami et al., 2020).

## 5 Experiment

### 5.1 Datasets

We evaluate our method on three well-known benchmarks, which are FB15k-237 (Toutanova and Chen, 2015), WN18RR (Tim Dettmers and Riedel, 2018), and YAGO3-10 (Mahdisoltani and Suchanek, 2015).

**FB15k-237** is a modified version of FB15k extracted from Freebase (K. Bollacker and Taylor, 2008), which excludes inverse relations to resolve a flaw with FB15k (Tim Dettmers and Riedel, 2018). The main relation patterns in FB15k-237 are asymmetry and composition.

**WN18RR** (Tim Dettmers and Riedel, 2018) is a subset of WN18 (Bordes et al., 2013) from WordNet (Miller, 1995), which retains most of the symmetric, asymmetric and compositional relations while removing the inversion relations.

**YAGO3-10** is a subset of YAGO (Suchanek et al., 2007), a dataset which integrates vocabulary definitions of WordNet with classification system of Wikipedia. The statistics of three datasets are shown in Table 2.

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

Table 2: Statistics of FB15k-237, WN18RR, and YAGO3-10 datasets.

### 5.2 Experimental settings

**Training details.** During optimization procedure, we additionally adopted the following techniques for obtaining better performance. First, when pre-processing datasets, we follow the data augmentation protocol in (Lacroix et al., 2018) by using reciprocal relations. Second, we utilized nuclear $p$-Norm regularization method proposed by (Lacroix et al., 2018). The reported result is the average result after three runnings.

**Evaluation protocol.** We evaluate the ALMP and baseline models on two widely used evaluation metrics: mean reciprocal rank (MRR), and top-$k$ Hit Ratio (Hit@$k$). For each valid triples $(h, r, t)$ in the test set, we replace either $h$ or $t$ with every other entities in the dataset to create corrupted triples in the link prediction task. Following previous work (Bordes et al., 2013; Tim Dettmers and Riedel, 2018; Nathani et al., 2019), all the models are evaluated in a *filtered* setting, i.e, corrupt triples that appear in training, validation, or test sets are removed during ranking. The valid triple and filtered corrupted triples are ranked in ascending order based on their prediction scores. Higher MRR or Hit@$k$ indicate better performance.

**Model setting.** We simply denote the model with the classic ALMP framework as the ALMP model. The attention module of ALMP adopts an element-wise scaled dot-product attention, which is similar to (Chami et al., 2020). The aggregation module of ALMP adopts a simple Hadamard product and a summation over all dimensions.

**Baselines.** We compared ALMP with a number of representative baselines, which are TransE, ComplEx, RotatE, Rot-Pro (Song et al., 2021), ATTE, BoxE (Abboud et al., 2020), TuckER (Balazevic et al., 2019), and PairRE (Chao et al., 2021). ATTE is the variant of ATTH (Chami et al., 2020) on euclidean space, which integrated two geometric operation: rotation and refection. We choose ATTE instead of ATTH to focus on the knowledge graph embedding models on euclidean space.

**ALMP variants.** We further build a set of ALMP variants by modifying a specific module in ALMP for the ablation study afterwards. The illustration of ALMP variants is as follows.

- **DLMP** (Disentangled Learning for Multi-relation Patterns) is a variant whose entity embeddings under reflection, rotation, and projection are disentangled from each other.

- **ALMP*** utilizes non-element-wise attention instead of the element-wise attention mechanism in ALMP.

| | FB15k-237 | | | | WN18RR | | | | YAGO3-10 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MRR | Hit@1 | Hit@3 | Hit@10 | MRR | Hit@1 | Hit@3 | Hit@10 | MRR | Hit@1 | Hit@3 | Hit@10 |
| TransE [†] | .294 | - | - | .465 | .226 | - | - | .501 | - | - | - | - |
| ComplEx [†] | .247 | .158 | .275 | .428 | .44 | .41 | .46 | .51 | .36 | .26 | .40 | .55 |
| RotatE [†] | .338 | .241 | .375 | .533 | .476 | .428 | .492 | .571 | .495 | .402 | .550 | .670 |
| Rot-Pro | .344 | .246 | .383 | .540 | .457 | .397 | .482 | .577 | .542 | .443 | .596 | .699 |
| ATTE | .351 | .255 | .386 | .543 | .489 | .443 | .504 | .577 | .525 | .440 | .574 | .680 |
| BoxE | .337 | .238 | .347 | .538 | .451 | .400 | .472 | .541 | .560 | .484 | .608 | .691 |
| TuckER | **.358** | **.266** | **.394** | .544 | .470 | .443 | .482 | .526 | - | - | - | - |
| PairRE | .351 | .256 | .387 | .544 | - | - | - | - | - | - | - | - |
| DLMP | .348 | .253 | .384 | .543 | **.498** | **.451** | **.516** | **.589** | .539 | .451 | .604 | .696 |
| ALMP$^-$ | .347 | .250 | .386 | .542 | .454 | .399 | .473 | .577 | .515 | .439 | .558 | .656 |
| ALMP* | .353 | .257 | **.390** | **.548** | .494 | .448 | .511 | .585 | .542 | .462 | .586 | .688 |
| ALMP | **.355** | **.260** | .319 | **.548** | .488 | .439 | .506 | .586 | **.566** | **.489** | **.612** | **.702** |

Table 3: Link prediction results on FB15k-237, WN18RR and YAGO3-10. Results of models with [†] are taken from (Sun et al., 2019). The result of ATTE is reproduced by us with suggested hyper-parameters. Other results are taken from the original paper of corresponding model.

- **ALMP$^-$** is another variant of ALMP with no additional step of translation transformation.

**Hyper-parameter settings.** We train ALMP and its variants using a grid search of hyper-parameters: embedding dimensions in $\{400, 500, 600\}$; learning rate in $\{1e^{-5}, 1e^{-4}, 5e^{-4}\}$; batch size in $\{512, 1024, 2048\}$; number of negative sampling in $\{0, 50, 100, 200\}$.

## 5.3 Main results

The experimental results on three datasets are reported in Table 3. We can see that ALMP with its variants outperforms most the baseline models across all common evaluation metrics, which empirically show the effectiveness of integrating affine transformations with attention to model complex interactions among relational patterns.

Furthermore, the performance gains over Rot-Pro could confirm the stronger expressiveness of integrated relational inductive bias than single bias. In other words, although relation patterns can be theoretically modeled separately by Rot-Pro, integrating them can indeed gain better performance, which coincides with our suppose of partial and global relation patterns. Meanwhile, the performance gain over AttE illustrates that integrating more forms of transformations gains better performance on expressiveness.

## 5.4 Ablation study on ALMP variants

According to the results of three variants of ALMP, we could draw some experimental conclusions of ALMP according to their difference.

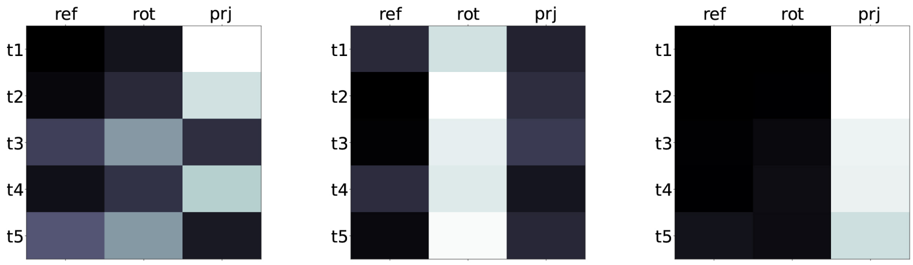
First, we could find that the performance of ALMP$^-$ is lower than other variants, which demonstrates that the effectiveness of using translation transformation as relational inductive biases. Second, DLMP outperforms other variants on WN18RR and have a reasonable performance on the other two datasets, which shows disentangled learning is also an effective approach for integrating various relation patterns for certain knowledge graphs. Furthermore, the link prediction result on ALMP* is similar to that on ALMP, where ALMP* shows more robust result across the three datasets, while ALMP show better results on both FB15k-237 and YAGO3-10.

In summary, the variants of ALMP show overall strong capability for knowledge graph completion. Also, for different knowledge graphs, it is feasible to fine tune the result with a specific variant.

## 5.5 Case study

For case study, we evaluate the MRR result for specific relations on WN18RR and YAGO3-10.

The relations we select basically contains the common global or partial relation patterns mentioned above. We compare the case study result with Rot-Pro, which is proved that could model the five relation patterns separately. The result is summarized in Table 4. The relations are selected manually with multiple relation properties. The case study result reflects that ALMP shows superior link prediction capability in modelling multiple relation patterns compared with KG embedding methods that model the relational properties separately.




(a) isConnectedTo (b) isLocateIn (c) isAfflictedTo

Figure 3: The attention visualization result of ALMP. Figure (a), (b), (c) represent the results of five triples randomly selected on the test set with different relations. Darker block represents lower attention value distributed to each transformation.

| Relation | Rot-Pro | ALMP |
|---|---|---|
| isConnectedTo | 0.405 | **0.423** |
| isLocatedIn | 0.297 | **0.335** |
| isAffiliatedTo | 0.664 | **0.725** |
| playsFor | 0.630 | **0.667** |
| hypernym | 0.150 | **0.199** |
| derivationally_related_form | **0.958** | 0.956 |
| instance_hypernym | 0.325 | **0.389** |
| also_see | **0.627** | 0.618 |
| member_meronym | 0.256 | **0.266** |
| synset_domain_topic_of | 0.347 | **0.411** |
| has_part | 0.197 | **0.198** |
| member_of_domain_usage | 0.308 | **0.382** |
| member_of_domain_region | 0.251 | **0.402** |

Table 4: Comparison of MRR on Rot-Pro and ALMP for typical relations on WN18RR and YAGO3-10.

## 5.6 Attention distribution validation

Having confirmed that ALMP could indeed gain better prediction performance on relations of multiple patterns, we seek to explore that for a specific relation, how the attention value is distributed to each geometric operation. In other words, we would figure out that, for two specific entities with a partial relational property annotated by humans, does the learned attention distributions accurately reflect it? To this end, we select five triples for three relations in YAGO3-10 respectively and draw the attention visualization graph in Figure 3.

We could find that there is obvious difference of attention distribution. For relation *isConnectedTo*, the attention distributed to three geometric operation is almost equal. It is may caused by that *isConnectedTo* is a relation with partial pattern of both symmetry, which can be learnt by both reflection and rotation, and transitivity, which can be modeled by projection. For relation *isLocatedIn*, the model tends to focus more attention on the rotation transformation. The possible reason might be that *isLocatedIn* is a relation with both global transitivity and asymmetry patterns, and asymmetry pattern could not be modeled by projection or reflection. As for relation *isAfflictedTo*, it is a relation with partial transitivity, hence the model prone to pay attention to the projection transformation.

## 5.7 Time and space complexity

The limitation of ALMP is that it consumes more computation, because each relation is represented as four transformations. However, the time complexity is still $\mathcal{O}(n)$. Due to the data parallel computing technique, the time efficiency is also similar to other models. We compare the time and space complexity between ALMP and ATTE (Chami et al., 2020) in Table 5.

| | Space | Time | $T$ (ms) |
|---|---|---|---|
| ATTE | $\mathcal{O}(n)\{\Theta_e, \Theta_r(\theta_{r_{1,2}})\}$ | $\mathcal{O}(n)$ | 0.254 |
| ALMP | $\mathcal{O}(n)\{\Theta_e, \Theta_r(\theta_{r_{1,2,3}}, a, b)\}$ | $\mathcal{O}(n)$ | 0.256 |

Table 5: The time and space complexities where $T$ represents the average predicting time per test triple with the same testing batch size.

# 6 Conclusion

In this paper, we proposed ALMP, a generic framework for knowledge graph embedding. ALMP could help handle the problem of the multiple relational properties by utilizing attention mechanism to integrating common affine transformation methods. On common KG embedding benchmarks, ALMP and its variants show effectiveness on link prediction task. According to the attention analysis and case study, ALMP is capable of capturing multiple relational properties.

## References


Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. BoxE: A box embedding model for knowledge base completion.

X. He J. Gao B. Yang, W.-t. Yih and L. Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*, pages 1–13.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Peter W Battaglia, Jessica B Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, and R. Faulkner. 2018. Relational inductive biases, deep learning, and graph networks. *CoRR*.

Yoshua Bengio. 2019. The consciousness prior.

Yoshua Bengio, Yann Lecun, and Geoffrey Hinton. 2021. Deep learning for ai. *Commun. ACM*, 64(7):58–65.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*, pages 2787–2795.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914. Association for Computational Linguistics.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. 2021. PairRE: Knowledge graph embeddings via paired relation vectors. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4360–4369, Online. Association for Computational Linguistics.

P. Paritosh T. Sturge K. Bollacker, C. Evans and J. Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, page 1247–1250.

Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. 2020. Representation learning for dynamic graphs: A survey.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion.

Biega J. Mahdisoltani, F. and F. M. Suchanek. 2015. Yago3: A knowledge base from multilingual wikipedias. *Proceedings of CIDR 2015*.

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

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723.

Tengwei Song, Jie Luo, and Lei Huang. 2021. Rotpro: Modeling transitivity by projection in knowledge graph embedding. In *Proceedings of the Thirty-Fifth Annual Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, page 697–706.

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

S. Riedel E. Gaussier T. Trouillon, J. Welbl and G. Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of 33rd Int. Conf. Mach. Learn*, page 2071–2080.

Pontus Stenetorp Tim Dettmers, Pasquale Minervini and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, page 57–66.

R.J. Valenza. 2012. *Linear Algebra: An Introduction to Abstract Mathematics*. Undergraduate Texts in Mathematics. Springer New York.

R. Wang, B. Li, S. Hu, W. Du, and M. Zhang. 2020. Knowledge graph embedding via graph attenuated attention networks. *IEEE Access*, 8:5212–5224.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI Conference on Artificial Intelligence*.

Maosong Sun Yang Liu Yankai Lin, Zhiyuan Liu and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2181–2187.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019a. Quaternion knowledge graph embedding. *Advances in Neural Information Processing Systems*, page 2731–2741.

Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019b. Iteratively learning embeddings and rules for knowledge graph reasoning.