MEIM: Multi-partition Embedding Interaction Beyond Block Term Format for Efficient and Expressive Link Prediction

Hung-Nghiep Tran[1] , **Atsuhiro Takasu**[1,2]

[1]National Institute of Informatics, Tokyo, Japan
[2]The Graduate University for Advanced Studies, SOKENDAI, Tokyo, Japan
{nghiepth, takasu}@nii.ac.jp

Abstract

Knowledge graph embedding aims to predict the missing relations between entities in knowledge graphs. Tensor-decomposition-based models, such as ComplEx, provide a good trade-off between efficiency and expressiveness, that is crucial because of the large size of real world knowledge graphs. The recent multi-partition embedding interaction (MEI) model subsumes these models by using the block term tensor format and provides a systematic solution for the trade-off. However, MEI has several drawbacks, some of which carried from its subsumed tensor-decomposition-based models. In this paper, we address these drawbacks and introduce the Multi-partition Embedding Interaction iMproved beyond block term format (MEIM) model, with independent core tensor for ensemble effects and soft orthogonality for max-rank mapping, in addition to multi-partition embedding. MEIM improves expressiveness while still being highly efficient, helping it to outperform strong baselines and achieve state-of-the-art results on difficult link prediction benchmarks using fairly small embedding sizes. The source code is released at https://github.com/tranhungnghiep/MEIM.

1 Introduction

Knowledge graphs are used to represent relational information between entities. There are large real world knowledge graphs such as YAGO [Mahdisoltani *et al.*, 2015] containing millions of entities. These knowledge graphs and their representations can be used in artificial intelligent applications such as semantic queries and question answering [Tran and Takasu, 2019b] [Tran, 2020].

Knowledge graph embedding aims to predict the missing relations between entities in knowledge graphs. They usually represents a triple (h, t, r) as embeddings and use a score function to compute its matching score $\mathcal{S}(h, t, r)$. The score function defines the *interaction mechanism* such as bilinear map between the embeddings and the *interaction pattern* specifying how embedding entries interact with each other.

On large real world knowledge graphs, a good tradeoff between efficiency and expressiveness is crucial. Tensor-decomposition-based models usually provide good trade-off by designing special interaction mechanisms with sparse and expressive interaction patterns. They have been subsumed by the recent multi-partition embedding interaction (MEI) model [Tran and Takasu, 2020], that divides the embedding vector into multiple partitions for sparsity, automatically learns the local interaction patterns on each partition for expressiveness, then combines the local scores to get the full interaction score. The trade-off between efficiency and expressiveness can be systematically controlled by changing the partition size and learning the interaction patterns through the core tensor of the block term tensor format.

However, by looking from two perspectives beyond the scope of block term format, that is, ensemble boosting effects and max-rank relational mapping, we identify two drawbacks of MEI and related tensor-decomposition-based models. First, using one core tensor for all partitions leads to similar local interactions, that may harm the ensemble effects of the full interaction in MEI. Second, the relational mapping matrices in MEI are generated from relatively small relation embeddings, that may make the model prone to degenerate.

In this paper, we propose the novel Multi-partition Embedding Interaction iMproved beyond block term format (MEIM) model with two techniques, namely independent core tensor to improve ensemble effects and max-rank mapping by soft orthogonality to improve model expressiveness. This introduces new aspects to tensor-decomposition-based models exploiting ensemble boosting effects and max-rank relational mapping, in addition to multi-partition embedding by MEI. MEIM improves expressiveness while still being highly efficient, helping it to outperform strong baselines and achieve new state-of-the-art results on difficult link prediction benchmarks using fairly small embedding sizes.

In general, our contributions include the following.

- We propose MEIM, a novel tensor-decomposition-based model with independent core tensor and max-rank mapping by soft orthogonality to improve expressiveness.

- We extensively experiment to show that the proposed model is highly efficient and expressive, and achieves state-of-the-art results.

- We analyze the proposed model to clarify its characteristics and demonstrate the critical advantages of its soft orthogonality compared to previous models.
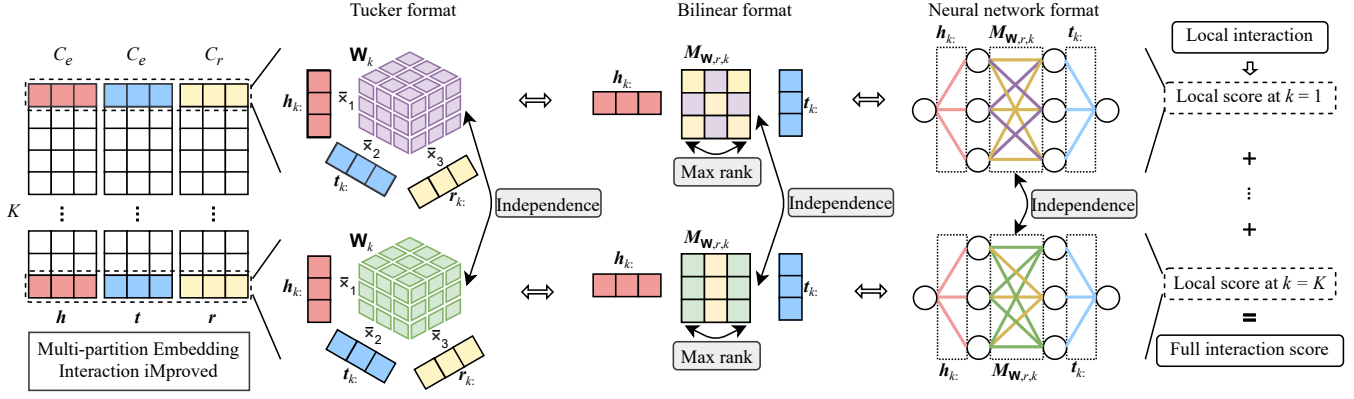
Figure 1: MEIM architecture: multi-partition embedding interaction improved with independent core tensors and max-rank mapping matrices. The new aspects are noted by the grey boxes.

2 Background

2.1 Notations and Definitions

A knowledge graph is a collection of triples $\mathcal{D}$, with each triple denoted as a tuple (h, t, r), where h and t are head and tail entities in the entity set $\mathcal{E}$ and r belongs to the relation set $\mathcal{R}$. A knowledge graph can also be represented by a third-order binary *data tensor* $\mathbf{G} \in \{0, 1\}^{|\mathcal{E}| \times |\mathcal{E}| \times |\mathcal{R}|}$, where each entry $g_{htr} = 1 \Leftrightarrow (h, t, r)$ exists in $\mathcal{D}$.

The *contextual link prediction* task, also called link prediction, aims to predict the connection between two entities given a relation as the context.

2.2 Tensor-Decomposition-based Knowledge Graph Embedding Models

There are several knowledge graph embedding models that adapt tensor formats to represent the knowledge graph data tensor and adapt tensor decomposition methods to solve the link prediction task. This approach has led to some of the best models in terms of efficiency and expressiveness.

RESCAL [Nickel *et al.*, 2011] is an early model that adapts Tucker2 decomposition to compute the embedding interaction score by a bilinear map

$$\mathcal{S}(h, t, r) = \boldsymbol{h}^\top \boldsymbol{M}_r \boldsymbol{t}, \qquad (1)$$

where $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{R}^D$, and $\boldsymbol{M}_r \in \mathbb{R}^{D \times D}$ are the embeddings of h, t, and r, respectively, with D being the embedding size. RESCAL is expressive, but the mapping matrix $\boldsymbol{M}_r$ grows quadratically with embedding size, making the model expensive and prone to overfitting.

The most simple model is DistMult [Yang *et al.*, 2015], that uses a sparse diagonal mapping matrix $\boldsymbol{M}_r = \mathrm{diag}(\boldsymbol{r})$, where $\boldsymbol{r} \in \mathbb{R}^D$ is the relation embedding vector. It is an adaptation of CP decomposition [Kolda and Bader, 2009], with the score function written as a trilinear product

$$\mathcal{S}(h, t, r) = \boldsymbol{h}^\top \mathrm{diag}(\boldsymbol{r}) \boldsymbol{t} = \sum_i h_i t_i r_i = \langle \boldsymbol{h}, \boldsymbol{t}, \boldsymbol{r} \rangle. \quad (2)$$

Recent models aim to be more expressive than DistMult but still efficient by designing new special interaction mechanisms between the embeddings. It has been shown that these models are equivalent to bilinear model with sparse block-diagonal mapping matrix $\boldsymbol{M}_r$,

$$\boldsymbol{M}_r = \begin{bmatrix} \boldsymbol{M}_{r,1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{M}_{r,K} \end{bmatrix}, \qquad (3)$$

where the matrix block $\boldsymbol{M}_{r,k}$ have a special interaction pattern resulted from the specific interaction mechanism [Tran and Takasu, 2020]. For example, ComplEx [Trouillon *et al.*, 2016] uses complex-valued trilinear product, resulting in the 2-dimensional rotation–scaling pattern on complex plane,

$$\boldsymbol{M}_{r,k} = \begin{bmatrix} \mathrm{real}(r)_k & -\mathrm{imaginary}(r)_k \\ \mathrm{imaginary}(r)_k & \mathrm{real}(r)_k \end{bmatrix}. \qquad (4)$$

Sparsity and the interaction pattern are crucial concepts in previous tensor-decomposition-based models. They are generalized by the recent multi-partition embedding interaction (MEI) model [Tran and Takasu, 2020], that divides the embedding vector into multiple partitions, learns the local interaction patterns on each partition, then sums the local scores to get the full interaction score. The trade-off between efficiency and expressiveness can be systematically controlled by changing the partition size and learning the interaction patterns through the core tensor of the block term tensor format.

3 Multi-partition Embedding Interaction iMproved Beyond Block Term Format

In this section, we propose the MEIM model that introduces new aspects to tensor-decomposition-based models exploiting *ensemble boosting effects* and *max-rank relational mapping*, in addition to multi-partition embedding by MEI, as illustrated in Figure 1.

3.1 The Score Function

Inspired by the generality of MEI, we adopt the multi-partition embedding representation, with Tucker format [Tucker, 1966] for local interaction and block term format [De Lathauwer, 2008] for full interaction. Following convention of MEI, triple (h, t, r) provides the embedding vectors

$\boldsymbol{h}, \boldsymbol{t} \in \mathbb{R}^{D_e}$, and $\boldsymbol{r} \in \mathbb{R}^{D_r}$ that are treated as multi-partition embedding matrices $\boldsymbol{H}, \boldsymbol{T} \in \mathbb{R}^{K \times C_e}$, and $\boldsymbol{R} \in \mathbb{R}^{K \times C_r}$, respectively. The score function of MEIM is written as

$$\mathcal{S}(h, t, r; \boldsymbol{\theta}) = \sum_{k=1}^{K} \mathcal{S}_k(h, t, r; \boldsymbol{\theta}) \tag{5}$$

$$= \sum_{k=1}^{K} \mathbf{W}_k \bar{\times}_1 \boldsymbol{h}_{k:} \bar{\times}_2 \boldsymbol{t}_{k:} \bar{\times}_3 \boldsymbol{r}_{k:}, \tag{6}$$

$$= \sum_{k=1}^{K} \boldsymbol{h}_{k:}^{\top} (\mathbf{W}_k \bar{\times}_3 \boldsymbol{r}_{k:}) \boldsymbol{t}_{k:} \tag{7}$$

$$= \sum_{k=1}^{K} \boldsymbol{h}_{k:}^{\top} \boldsymbol{M}_{\mathbf{W}, r, k} \boldsymbol{t}_{k:}, \tag{8}$$

where $\boldsymbol{h}_{k:}, \boldsymbol{t}_{k:}$, and $\boldsymbol{r}_{k:}$ are the embedding partitions k; $\mathbf{W}_k \in \mathbb{R}^{C_e \times C_e \times C_r}$ is the core tensor at partition k; $\bar{\times}_n$ denotes the n-mode tensor product with a vector; and $\boldsymbol{M}_{\mathbf{W}, r, k} \in \mathbb{R}^{C_e \times C_e}$ is the bilinear mapping matrix.

Note that Eq. 5 shows the sum of local interaction scores $\mathcal{S}_k(h, t, r; \boldsymbol{\theta})$. Eq. 6 shows the block term format. Eq. 8 shows the bilinear format with block-diagonal mapping matrix, in which each block $\boldsymbol{M}_{\mathbf{W}, r, k}$ is generated by $\mathbf{W}_k \bar{\times}_3 \boldsymbol{r}_{k:}$. Eq. 8 can be seen as a dynamic linear neural network [Tran and Takasu, 2020], where the hidden layer $\boldsymbol{M}_{\mathbf{W}, r, k}$ is generated by another hyper neural network [Ha *et al.*, 2016] with the core tensor $\mathbf{W}_k$ as its weights and $\boldsymbol{r}_{k:}$ as its input.

The main novelties of MEIM are in better parameterization of the core tensor and mapping matrix, by looking from two perspectives beyond the scope of block term format, that is, ensemble boosting effects and max-rank relational mapping.

3.2 Core Tensor for Ensemble Boosting

Technically, the score function in Eq. 5 can be seen as an ensemble system of K local interactions by summing their scores. This works in a similar manner to gradient boosting [Mason *et al.*, 1999] because the full interaction score is computed at training time and gradients are back-propagated to optimize all local interactions together, enabling them to implicitly minimize the residual error of each other.

The MEI model briefly mentioned this perspective but did not actively exploit the ensemble boosting effects. MEI used only one core tensor for all partitions by enforcing the following *shared core tensor* constraint:

$$\mathbf{W}_1 = \mathbf{W}_2 = \cdots = \mathbf{W}_K = \mathbf{W}, \tag{9}$$

where $\mathbf{W} \in \mathbb{R}^{C \times C \times C}$ is the learned shared core tensor, although block term format may have different core tensors. MEI learns one interaction pattern for all partitions, similar to that of other tensor-decomposition-based models such as ComplEx. However, this may be harmful to the performance of an ensemble system because of the similar local interaction patterns instead of combining different patterns. To exploit the ensemble boosting effects, it is crucial to promote independence and difference between the local interactions.

Independent Core Tensor Parameterization

There are several potential ways to resolving the problem of independence and difference between the local interactions. First, we may enforce the predicted scores to be different:

$$\forall k, l \in \{1, \ldots, K\}, k \neq l : \mathcal{S}_k(\cdot, \cdot, \cdot) \neq \mathcal{S}_l(\cdot, \cdot, \cdot). \tag{10}$$

However, because this forces the predicted scores to be not the same, the local interactions cannot be correct at the same

time. Thus, this is an overstrong constraint that may cause more harm than good. Another way is to explicitly enforce different parameters of the local interaction functions, such as the core tensors:

$$\forall k, l \in \{1, \ldots, K\}, k \neq l : \mathbf{W}_k \neq \mathbf{W}_l. \tag{11}$$

However, such explicit difference constraints do not guarantee different interactions because of symmetric swapping or scaling reparameterizations of the core tensors and the embeddings. For example, scaling the core tensor by x and the corresponding embedding by $1/x$ do not change the score.

Therefore, in this paper, we do not use difference constraints, but instead independence constraints, denoted as:

$$\mathbf{W}_1 \perp \mathbf{W}_2 \perp \cdots \perp \mathbf{W}_K. \tag{12}$$

For simplicity, we let the model implicitly learn *independent core tensors* from data automatically by removing the constraint in Eq. 9. Using independent core tensors enables the model to learn independent and possibly different local interactions to improve the ensemble boosting effects. More advanced methods based on further studying the ensemble boosting effects are left for future work.

Note that MEI is easy to implement in common deep learning frameworks by using only matrix product. However, independent core tensor makes the multi-partition tensor product in MEIM more difficult to implement efficiently. To resolve this problem, we stack together K independent core tensors as a fourth-order tensor:

$$\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_K], \tag{13}$$

where $\mathbf{W} \in \mathbb{R}^{K \times C \times C \times C}$ is used for the Einstein sum notation technique to create an efficient implementation.

3.3 Max-Rank Relational Mapping

MEI is a *contextual link prediction* model with two components, the multi-partition relation-based contextual mapping and the simple dot product matching. The MEI model used a linear mapping where $\boldsymbol{M}_{\mathbf{W}, r, k}$ can be any $C_e \times C_e$ square matrix. However, when some columns of $\boldsymbol{M}_{\mathbf{W}, r, k}$ are dependent, it becomes a singular matrix with rank smaller than C_e. Such singular matrix would map the head embedding to a subspace with lower dimension, thus reduce the embedding space effective size. It may make the score function degenerate to always produces score 0 for some entities, thus, drastically reduces the expressiveness of the model. This may become a critical issue in MEI because $\boldsymbol{M}_{\mathbf{W}, r, k}$ is generated from a relatively small relation embedding partition $\boldsymbol{r}_{k:}$ as shown in Eq. 8. To resolve this problem, the mapping matrices $\boldsymbol{M}_{\mathbf{W}, r, k}$ need to have max rank, that may not always be full rank C_e, but should be as large as possible.

Max-Rank Mapping Matrix by Soft Orthogonality

There are several different types of full rank matrices. A particularly interesting case is the orthogonal matrices, that act as the rotational transformations when the determinant is 1 and an additional reflection across the origin when the determinant is -1. These matrices have linearly independent column vectors with unit Frobenius norms, written as the constraints:

$$\forall k \in \{1, \ldots, K\} : \boldsymbol{M}_{\mathbf{W}, r, k}^{\top} \boldsymbol{M}_{\mathbf{W}, r, k} = \boldsymbol{I}, \tag{14}$$

where $\boldsymbol{I} \in \mathbb{R}^{C_e \times C_e}$ is the identity matrix. In addition, note that the mapping matrices are generated by the core tensors and the relation embedding vectors. To reduce the search space and make optimization easier, we also constrain each relation partition $\boldsymbol{r}_{k:}$ to have unit norm:

$$\forall k \in \{1, \ldots, K\} : \boldsymbol{r}_{k:}^\top \boldsymbol{r}_{k:} = 1. \quad (15)$$

Because of the way $\boldsymbol{M}_{\mathsf{W},r,k}$ are generated, it may be difficult to enforce the mapping matrices to be strictly orthogonal. Even if it is possible, it may not always be preferable because the main objective is link prediction. Geometrically speaking, the orthogonal matrices form the Stiefel manifold [Sokół and Park, 2020], we need to make sure that the mapping matrices stay as close to this manifold as possible without hurting the link prediction accuracy. Using Lagrangian relaxation, the hard constraints in Eq. 14 and Eq. 15 can be converted to the following soft orthogonality loss term:

$$\begin{aligned} \mathcal{L}_{\mathrm{ortho}} = \lambda_{\mathrm{ortho}} \big(& \textstyle\sum_{k=1}^K ||\boldsymbol{M}_{\mathsf{W},r,k}^\top \boldsymbol{M}_{\mathsf{W},r,k} - \boldsymbol{I}||_2^2. \\ & + \lambda_{\mathrm{unitnorm}} \textstyle\sum_{k=1}^K |\boldsymbol{r}_{k:}^\top \boldsymbol{r}_{k:} - 1|^p \big), \quad (16) \end{aligned}$$

where λ_{ortho}, $\lambda_{\mathrm{unitnorm}}$, and p are hyperparameters. To penalize large differences from unit norm, $p = 3$ is used. When λ_{ortho} is small, the model tends to learn linear mappings. When it is large, the model tends to learn strictly orthogonal mappings.

The soft orthogonality loss enables the model to balance between orthogonality and link prediction. Minimizing this loss effectively influences the column vectors to be orthogonal and have unit norms, pushes the matrix $\boldsymbol{M}_{\mathsf{W},r,k}$ close to the Stiefel manifold and maximizes its rank in balance with the link prediction objective.

3.4 Connections to Previous Models

Compared to the MEI model, MEIM inherits its key benefit, that is, the systematic trade-off between efficiency and expressiveness by controlling the partition size and learning the interaction patterns. About computational cost, the number of parameters in MEIM is similar to that of MEI, except that the independent core tensors use $KC_e^2 C_r$ instead of $C_e^2 C_r$ parameters, but this becomes less relevant when the number of entities and relations are large, and MEIM is as fast as MEI by using Einstein sum notation technique for implementation of independent core tensor product.

On the other hand, MEIM introduces new parameterization techniques to address some major drawbacks of MEI. With independent core tensor, MEIM can learn independent local interaction patterns to improve ensemble effects. With soft orthogonality, MEIM balances the orthogonality and link prediction objectives to learn max-rank mapping that improves expressiveness of the model.

Note that, coincidentally, some knowledge graph embedding models also try to use orthogonal mapping, not to maximize the mapping matrix rank but to model some relational patterns, such as symmetry, inversion, and composition [Sun *et al.*, 2019]. There are many different ways to design and use orthogonal mapping in knowledge graph embedding. For example, RotatE [Sun *et al.*, 2019] uses rotation in the complex plane, QuatE [Zhang *et al.*, 2019] uses rotation in the quaternion space, GC-OTE [Tang *et al.*, 2020] uses rotation by the

Gram Schmidt process, and RotH [Chami *et al.*, 2020] uses rotation by the Givens matrix. However, there are some important differences to our method. First, all of them use rigid orthogonality where the mapping matrix is strictly orthogonal. This may be suboptimal because orthogonal mapping may not always be useful for link prediction, and even when it is useful, strictly orthogonal mapping may reduce the potential result. In contrast, we approach the problem from the perspective of maximizing the mapping matrix rank, thus, we use soft orthogonality as a means instead of rigid orthogonality. Soft orthogonality can avoid the above problems because we can choose how to balance the orthogonality and the link prediction objectives for each dataset, as we will show this is crucial for knowledge graph embedding in experiments. Second, MEIM has a totally different model architecture compared to previous orthogonal models, because the mapping matrix in MEIM is not learned directly but generated. Finally, MEIM is a semantic matching model, with the matching score computed by inner product, whereas other orthogonal models are translation-based model, in which the score is computed by distance.

4 Loss Function and Learning

The model is learned via the link prediction task that can be modeled as a multi-class classification problem. There are two directions, classifying correct tail entity t among all entities given a (h, r) pair, and classifying correct head entity h among all entities given a (t, r) pair. Following recent practice, we use the *1-vs-all* and *k-vs-all* sampling methods with the softmax cross-entropy loss function [Dettmers *et al.*, 2018] [Ruffinelli *et al.*, 2020].

First, using *1-vs-all* and *k-vs-all* sampling, we define the groundtruth categorical distributions over all entities $\mathcal{E}$ given (h, r) and given (t, r), denoted $\hat{p}_{hir}$ and $\hat{p}_{htr}$, respectively. Second, we compute the corresponding predicted distributions p_{hir} and p_{htr} using the softmax function on the matching scores. The link prediction loss is the cross-entropy summed over training data $\mathcal{D}$:

$$\begin{aligned} \mathcal{L}_{\mathrm{link\_prediction}} = \ -\textstyle\sum_{(h,t,r)\in\mathcal{D}} \big(& \textstyle\sum_{i\in\mathcal{E}} \hat{p}_{hir} \log p_{hir} \\ & + \textstyle\sum_{h\in\mathcal{E}} \hat{p}_{htr} \log p_{htr} \big). \quad (17) \end{aligned}$$

The final loss function is the sum of link prediction loss and soft orthogonality loss:

$$\mathcal{L} = \mathcal{L}_{\mathrm{link\_prediction}} + \mathcal{L}_{\mathrm{ortho}}, \quad (18)$$

with λ_{ortho} tuned to balance the two loss terms. For regularization, we only used dropout [Srivastava *et al.*, 2014] and batch normalization [Ioffe and Szegedy, 2015]. All parameters in the model are learned in an end-to-end fashion by minimizing the loss using mini-batch stochastic gradient descent with Adam optimizer [Kingma and Ba, 2014].

We did not use some recent complemental techniques for training, regularization, and additional losses that can be used with existing methods including MEIM. Adding them to MEIM to further improve the result is left for future work.

Dataset	$|\mathcal{E}|$	$|\mathcal{R}|$	Train	Valid	Test	Avg. degree
WN18RR	40,943	11	86,835	3,034	3,134	2.12
FB15K-237	14,541	237	272,115	17,535	20,466	18.71
YAGO3-10	123,182	37	1,079,040	5,000	5,000	8.76

Table 1: Datasets statistics.

5 Experiments

5.1 Experimental Settings

Datasets. We use three standard benchmark datasets, as shown in Table 1. WN18RR [Dettmers *et al.*, 2018] is a subset of WordNet containing lexical information. FB15K-237 [Toutanova and Chen, 2015] is a subset of Freebase containing general facts. In addition, YAGO3-10 [Mahdisoltani *et al.*, 2015] is a large and very competitive dataset containing general facts from Wikipedia.

Evaluations. We evaluate on the link prediction task. For each true triple (h, t, r) in the test set, we replace h and t by every other entity to generate corrupted triples (h', t, r) and (h, t', r), respectively. The model tries to rank the true triple (h, t, r) before the corrupted triples based on the score $\mathcal{S}$. We compute MRR (mean reciprocal rank) and $H@k$ for $k \in \{1, 3, 10\}$ (how many triples correctly ranked in the top k) [Trouillon *et al.*, 2016]. The higher MRR and $H@k$ are, the better the model performs. Filtered metrics are used to avoid penalization when ranking other true triples before the current target triple [Bordes *et al.*, 2013].

Note that MEIM can solve other tasks by converting them to *contextual link prediction* task. For example, by defining the *alignment relation* r_a, we have the alignment triples of the form (e_1, e_2, r_a), that can be used directly in MEIM for solving entity alignment. These tasks are left for future work.

Implementations. The MEIM model is implemented as a neural network using PyTorch. Following MEI, we use embedding size $K = 3, C = 100$ on WN18RR and FB15K-237, and $K = 5, C = 100$ on YAGO3-10. They are equivalent in terms of model size to ComplEx with embedding size 190 on WN18RR, 250 on FB15K-237, and 270 on YAGO3-10, that are relatively small compared to most related work. By preliminary experiments, we use *k-vs-all* sampling on WN18RR and *1-vs-all* sampling on other datasets. We use batch size 1024 and learning rate 3e-3, with exponential decay 0.995 on YAGO3-10 and 0.99775 on other datasets. Dropout and batch normalization are used on the input and hidden layers, that is, on $\boldsymbol{h}$ and $\boldsymbol{h}^\top \boldsymbol{M}_r$, respectively. Other hyperparameters are tuned by grid search, with grids $[0, 0.75]$ and step size 0.01 for drop rates, $\{1, 1e\text{-}1, 1e\text{-}2, 1e\text{-}3, 1e\text{-}4, 1e\text{-}5, 0\}$ for λ_{ortho}, $\{1, 1e\text{-}1, 1e\text{-}2, 1e\text{-}3, 5e\text{-}4, 1e\text{-}4, 0\}$ for $\lambda_{\text{unitnorm}}$. The following values were found. On WN18RR, input drop rate is 0.71, hidden drop rate is 0.67, $\lambda_{\text{ortho}} = 1e\text{-}1$, $\lambda_{\text{unitnorm}} = 5e\text{-}4$. On FB15K-237, input drop rate is 0.66, hidden drop rate is 0.67, $\lambda_{\text{ortho}} = 0$, $\lambda_{\text{unitnorm}} = 0$. On YAGO3-10, input drop rate is 0.1, hidden drop rate is 0.15, $\lambda_{\text{ortho}} = 1e\text{-}3$, $\lambda_{\text{unitnorm}} = 0$. Hyperparameters were tuned to maximize the validation MRR.

Baselines. The main baselines are MEI and tensor-decomposition-based models such as ComplEx, RotatE, QuatE that we aim to improve. MEI results are reproduced

	WN18RR		FB15K-237		YAGO3-10	
	#params	MRR	#params	MRR	#params	MRR
TuckER	9.4 M	0.470	11.0 M	0.358	–	–
RotatE	41.0 M	0.476	14.8 M	0.338	61.6 M	0.495
ComplEx-N3	20.5 M	0.480	7.4 M	0.357	61.6 M	0.569
RotH	20.5 M	0.496	7.4 M	0.344	61.6 M	0.570
MEI	15.7 M	0.481	7.4 M	0.365	66.5 M	0.578
MEIM	15.3 M	**0.499**	7.4 M	**0.369**	66.6 M	**0.585**

Table 2: Parameter efficiency compared to popular baselines.

	ComplEx-N3		MEI		MEIM	
#params	MRR	H@10	MRR	H@10	MRR	H@10
1.2 M	0.404	0.439	0.448	0.500	0.460	0.525
3.8 M	0.453	0.507	0.476	0.544	0.488	0.557
6.5 M	0.464	0.528	0.481	0.547	0.494	0.569
15.3 M	0.473	0.550	0.481	0.544	0.499	0.574

Table 3: Model scalability on WN18RR validation set.

in this paper using comparable model sizes and recent training techniques that lead to better performance than previously reported. We also compare to translation-based models such as TransE, neural-network-based models such as ConvE, and recent models such as RotH. We tried to evaluate against popular and strong baselines with comparable settings and evaluation protocol for fair and informative comparisons. These exclude incomparable results that use extra data, excessively large model size, complemental techniques, or inappropriate evaluation protocol [Sun *et al.*, 2020]. Note that even at normal model size, MEIM can get state-of-the-art results.

5.2 Parameter Efficiency

We first compare the results and model sizes of MEIM to that of popular baselines as shown in Table 2. We can see that MEIM strongly outperforms these baselines while using roughly similar number of parameters.

In addition, Table 3 shows the scalability of MEIM and other baselines at different model sizes. MEIM outperforms the baselines at every model size and achieves the best result overall. This demonstrates that MEIM achieves a better balance between efficiency and expressiveness.

5.3 Main Results

The main results for link prediction are shown in Table 4. In general, MEIM achieves good result on all three datasets, including the very large and difficult YAGO3-10 dataset. Considering the most important and robust metric MRR, MEIM outperforms all baselines by a large margin on all datasets.

The strongest baselines on FB15K-237 and YAGO3-10 include the MEI model. This demonstrates that MEI was a strong model, but MEIM successfully addresses the drawbacks of MEI and its subsumed tensor-decomposition-based models to significantly improve the result on all datasets.

Another strong baseline is the RotH model on WN18RR, but the results on FB15K-237 are quite weak. We notice that most baselines usually have good results on one dataset and

	WN18RR				FB15K-237				YAGO3-10			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE [Bordes *et al.*, 2013] [‡]	0.222	0.031	–	0.524	0.310	0.217	–	0.497	0.501	0.406	–	0.674
TorusE [Ebisu and Ichise, 2019] [♯]	0.452	0.422	0.464	0.512	0.305	0.217	0.335	0.484	0.342	0.274	–	0.474
ConvE [Dettmers *et al.*, 2018] [♯]	0.43	0.40	0.44	0.52	0.325	0.237	0.356	0.501	0.488	0.399	–	0.658
InteractE [Vashishth *et al.*, 2020a]	0.463	0.430	–	0.528	0.354	0.263	–	0.535	0.541	0.462	–	0.687
CompGCN [Vashishth *et al.*, 2020b]	0.479	0.443	0.494	0.546	0.355	0.264	0.390	0.535	–	–	–	–
RAGAT [Liu *et al.*, 2021]	0.489	<u>0.452</u>	0.503	0.562	<u>0.365</u>	<u>0.273</u>	0.401	0.547	–	–	–	–
RESCAL [Nickel *et al.*, 2011] [†]	0.467	–	–	0.517	0.357	–	–	0.541	–	–	–	–
DistMult [Yang *et al.*, 2015] [‡]	0.465	0.432	–	0.532	0.313	0.224	–	0.490	0.501	0.413	–	0.661
ComplEx-N3 [Trouillon *et al.*, 2016] [†]	0.480	0.435	0.495	0.572	0.357	0.264	0.392	0.547	0.569	0.498	0.609	0.701
RotatE [Sun *et al.*, 2019]	0.476	0.428	0.492	0.571	0.338	0.241	0.375	0.533	0.495	0.402	0.550	0.670
QuatE [Zhang *et al.*, 2019]	0.488	0.438	0.508	0.582	0.348	0.248	0.382	0.550	–	–	–	–
GC-OTE [Tang *et al.*, 2020]	0.491	0.442	0.511	<u>0.583</u>	0.361	0.267	0.396	0.550	–	–	–	–
RotH [Chami *et al.*, 2020]	<u>0.496</u>	0.449	<u>0.514</u>	**0.586**	0.344	0.246	0.380	0.535	0.570	0.495	0.612	0.706
TuckER [Balažević *et al.*, 2019]	0.470	0.443	0.482	0.526	0.358	0.266	0.394	0.544	–	–	–	–
MEI [Tran and Takasu, 2020] [⋆]	0.481	0.444	0.496	0.551	<u>0.365</u>	0.271	<u>0.402</u>	<u>0.552</u>	<u>0.578</u>	<u>0.505</u>	<u>0.622</u>	<u>0.709</u>
MEIM	**0.499**	**0.458**	**0.518**	0.577	**0.369**	**0.274**	**0.406**	**0.557**	**0.585**	**0.514**	**0.625**	**0.716**

Table 4: Link prediction results on WN18RR, FB15K-237, and YAGO3-10. [†] are reported in [Ruffinelli *et al.*, 2020], [†] in [Chami *et al.*, 2020], [‡] in [Rossi *et al.*, 2021], [♯] YAGO3-10 in [Rossi *et al.*, 2021], [⋆] are reproduced here, other results are reported in their papers.

	Ablation	MRR	H@10
	Full model	**0.499**	**0.574**
WN18RR	without *Multi-partition Embedding*	0.492	0.572
	without *Independent Core Tensor*	0.498	0.573
	without *Soft Orthogonality*	0.483	0.550
	Full model	**0.375**	**0.562**
FB15K-237	without *Multi-partition Embedding*	0.372	0.554
	without *Independent Core Tensor*	0.370	0.554
	Full model	**0.582**	**0.710**
YAGO3-10	without *Multi-partition Embedding*	0.580	0.706
	without *Independent Core Tensor*	0.581	0.708
	without *Soft Orthogonality*	0.574	0.708

Table 5: Model ablation result on the validation sets.

	WN18RR		FB15K-237		YAGO3-10	
λ_{ortho}	MRR	H@10	MRR	H@10	MRR	H@10
0	0.483	0.550	**0.375**	**0.562**	0.574	0.708
1e-5	0.482	0.548	0.374	0.561	0.581	**0.711**
1e-4	0.483	0.553	0.371	0.556	0.581	0.709
1e-3	0.484	0.554	0.372	0.559	**0.582**	0.710
1e-2	0.497	**0.577**	0.362	0.546	**0.582**	0.710
1e-1	**0.499**	0.574	0.360	0.541	0.581	0.706
1e0	0.497	0.568	0.355	0.532	0.576	0.695

Table 6: The effects of soft orthogonality on the validation sets.

poor results on other datasets. This sets MEIM apart as it can achieve good results on all three datasets.

5.4 Analyses and Discussions

Now we look into the characteristics of the model to understand why it gives good result.

Ablation Study

Table 5 shows the main ablation results of MEIM, in which each feature is removed from the full model to see how much it contributes to the final results. For comparison, we also ablate the *multi-partition embedding* inherited from MEI.

In general, we can see that removing the features decreases the results, which confirms their effectiveness. In particular, the *independent core tensor* works well on all three datasets. It strongly contributes to the full model results on FB15K-237, and slightly but consistently contributes to the final results on WN18RR and YAGO3-10. The effects of *soft orthogonality* is more complicated. On WN18RR and YAGO3-10,

soft orthogonality is crucial to achieve the best results. However, it is harmful on FB15K-237 and was not included in the full model. This mixed effect guarantees further detailed investigation as follows.

The Effects of Soft Orthogonality

Table 6 shows the results when λ_{ortho} is changing from no orthogonality to strong orthogonality. We see that strong orthogonality works well on WN18RR, average orthogonality works well on YAGO3-10, but none orthogonality works on FB15K-237. These results empirically demonstrate the weakness of previous models compared to MEIM.

For example, models with rigid orthogonality such as QuatE and RotH get strong results on WN18RR but weak on FB15K-237, whereas models without orthogonality such as RESCAL, TuckER, and MEI get strong results on FB15K-237 but weak on WN18RR. In contrast, MEIM with soft orthogonality can get strong results on both WN18RR and FB15K-237. More importantly, the large and difficult YAGO3-10 dataset requires average orthogonality. Thus, only MEIM can get the best results on this dataset. This shows a crucial advantage of MEIM, because its soft orthogonality can be tuned to work optimally on each dataset.

6 Related Work

Knowledge graph embedding methods can be categorized based on how they compute the matching score.

Tensor-decomposition-based models adapt tensor representation formats such as CP format, Tucker format, and block term format to represent the knowledge graph data tensor [Kolda and Bader, 2009]. They make up most of recent state-of-the-art models such as ComplEx [Trouillon *et al.*, 2016], SimplE [Kazemi and Poole, 2018], TuckER [Balažević *et al.*, 2019], and MEI [Tran and Takasu, 2020]. They often achieve good trade-off between efficiency and expressiveness. Especially, the recent MEI model provides a systematic solution for this trade-off. However, they have some limitations affecting the ensemble effects and making them prone to degenerate, that we address in this paper.

Neural-network-based models use a neural network to compute the matching score, such as ConvE [Dettmers *et al.*, 2018] using convolutional neural networks, CompGCN [Vashishth *et al.*, 2020b] using graph convolutional networks. These models are generally more expensive but not always get better results than tensor-decomposition-based models.

Translation-based models use geometrical distance to compute the score, with relation embeddings act as the translation vectors, such as TransE [Bordes *et al.*, 2013]. These models are efficient and intuitive, but they have limitations in expressive power [Kazemi and Poole, 2018].

There are several ways to use orthogonality in knowledge graph embedding, such as RotatE [Sun *et al.*, 2019] using complex product, Quaternion [Tran and Takasu, 2019a] and QuatE [Zhang *et al.*, 2019] using quaternion product, GC-OTE [Tang *et al.*, 2020] using the Gram Schmidt process, and RotH [Chami *et al.*, 2020] using the Givens matrix. These models usually compute the score using distance, differently from MEIM. Moreover, they use rigid orthogonality that is harmful on some datasets and potentially suboptimal on other datasets compared to soft orthogonality.

7 Conclusion

In this paper, we introduce new aspects to tensor-decomposition-based models exploiting ensemble boosting effects and max-rank relational mapping, in addition to multi-partition embedding by MEI. We propose the MEIM model with two new techniques, namely independent core tensor to improve ensemble effects and max-rank mapping by soft orthogonality to improve expressiveness. MEIM achieves state-of-the-art link prediction results, including the large and difficult YAGO3-10 dataset, using fairly small embedding sizes. Moreover, we analyze and demonstrate the limitations of previous rigid orthogonality models and show that MEIM with soft orthogonality works well on multiple datasets.

For future work, it is promising to continue research on combining deep learning techniques and tensor decomposition. It is also interesting to use MEIM to solve tasks such as entity alignment and recommendation by converting them to contextual link prediction.

References

[Balažević *et al.*, 2019] Ivana Balažević, Carl Allen, and Timothy M. Hospedales. TuckER: Tensor Factorization for Knowledge Graph Completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 5185–5194, 2019.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-Relational Data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

[Chami *et al.*, 2020] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914, 2020.

[De Lathauwer, 2008] Lieven De Lathauwer. Decompositions of a Higher-Order Tensor in Block Terms—Part II: Definitions and Uniqueness. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1033–1066, 2008.

[Dettmers *et al.*, 2018] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 1811–1818, 2018.

[Ebisu and Ichise, 2019] Takuma Ebisu and Ryutaro Ichise. Generalized Translation-based Embedding of Knowledge Graph. *IEEE Transactions on Knowledge and Data Engineering*, 32(5):941–951, 2019.

[Ha *et al.*, 2016] David Ha, Andrew M. Dai, and Quoc V. Le. HyperNetworks. In *International Conference on Learning Representations*, page 18, 2016.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[Kazemi and Poole, 2018] Seyed Mehran Kazemi and David Poole. SimplE Embedding for Link Prediction in Knowledge Graphs. In *Advances in Neural Information Processing Systems*, pages 4289–4300, 2018.

[Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, page 15, 2014.

[Kolda and Bader, 2009] Tamara G. Kolda and Brett W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, 2009.

[Liu *et al.*, 2021] Xiyang Liu, Huobin Tan, Qinghong Chen, and Guangyan Lin. RAGAT: Relation Aware Graph Attention Network for Knowledge Graph Completion. *IEEE Access*, 9:20840–20849, 2021.

[Mahdisoltani *et al.*, 2015] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *Proceedings of the Conference on Innovative Data Systems Research*, 2015.

[Mason *et al.*, 1999] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Boosting Algorithms as Gradient Descent. In *Advances in Neural Information Processing Systems*, pages 512–518, 1999.

[Nickel *et al.*, 2011] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A Three-Way Model for Collective Learning on Multi-Relational Data. In *International Conference on Machine Learning*, pages 809–816, 2011.

[Rossi *et al.*, 2021] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(2):1–49, 2021.

[Ruffinelli *et al.*, 2020] Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You CAN Teach an Old Dog New Tricks! On Training Knowledge Graph Embeddings. In *International Conference on Learning Representations*, page 20, 2020.

[Sokół and Park, 2020] Piotr Aleksander Sokół and Il Memming Park. Information Geometry of Orthogonal Initializations and Training. In *International Conference on Learning Representations*, page 17, 2020.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[Sun *et al.*, 2019] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*, page 18, 2019.

[Sun *et al.*, 2020] Zhiqing Sun, Shikhar Vashishth, Soumya Sanyal, Partha Talukdar, and Yiming Yang. A Re-evaluation of Knowledge Graph Completion Methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5516–5522, 2020.

[Tang *et al.*, 2020] Yun Tang, Jing Huang, Guangtao Wang, Xiaodong He, and Bowen Zhou. Orthogonal Relation Transforms with Graph Context Modeling for Knowledge Graph Embedding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 2713–2722, 2020.

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

[Tran and Takasu, 2019a] Hung Nghiep Tran and Atsuhiro Takasu. Analyzing Knowledge Graph Embedding Methods from a Multi-Embedding Interaction Perspective. In *Proceedings of the Data Science for Industry 4.0 Workshop at EDBT/ICDT*, page 7, 2019.

[Tran and Takasu, 2019b] Hung Nghiep Tran and Atsuhiro Takasu. Exploring Scholarly Data by Semantic Query on Knowledge Graph Embedding Space. In *Proceedings of the 23rd International Conference on Theory and Practice of Digital Libraries*, pages 154–162, 2019.

[Tran and Takasu, 2020] Hung Nghiep Tran and Atsuhiro Takasu. Multi-Partition Embedding Interaction with Block Term Format for Knowledge Graph Completion. In *Proceedings of the European Conference on Artificial Intelligence*, pages 833–840, 2020.

[Tran, 2020] Hung Nghiep Tran. *Multi-Relational Embedding for Knowledge Graph Representation and Analysis*. PhD thesis, The Graduate University for Advanced Studies, SOKENDAI, Japan, 2020.

[Trouillon *et al.*, 2016] Theo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. Complex Embeddings for Simple Link Prediction. In *International Conference on Machine Learning*, pages 2071–2080, 2016.

[Tucker, 1966] Ledyard R Tucker. Some Mathematical Notes on Three-Mode Factor Analysis. *Psychometrika*, 31(3):279–311, 1966.

[Vashishth *et al.*, 2020a] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. InteractE: Improving Convolution-based Knowledge Graph Embeddings by Increasing Feature Interactions. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 3009–3016, 2020.

[Vashishth *et al.*, 2020b] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based Multi-Relational Graph Convolutional Networks. In *International Conference on Learning Representations*, page 16, 2020.

[Yang *et al.*, 2015] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *International Conference on Learning Representations*, page 12, 2015.

[Zhang *et al.*, 2019] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. Quaternion Knowledge Graph Embedding. In *Advances in Neural Information Processing Systems*, pages 2735–2745, 2019.