CONVOLUTIONAL COMPLEX KNOWLEDGE GRAPH EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we study the problem of learning continuous vector representations of knowledge graphs for predicting missing links. We present a new approach called CONEX, which infers missing links by leveraging the composition of a 2D convolution with a Hermitian inner product of complex-valued embedding vectors. We evaluate CONEX against state-of-the-art approaches on the WN18RR, FB15K-237, KINSHIP and UMLS benchmark datasets. Our experimental results show that CONEX achieves a performance superior to that of state-of-the-art approaches such as RotatE, QuatE and TuckER on the link prediction task on all datasets while requiring at least 8 times fewer parameters. We ensure the reproducibility of our results by providing an open-source implementation which includes the training, evaluation scripts along with pre-trained models at `https://github.com/conex-kge/ConEx`.

1 INTRODUCTION

Knowledge Graphs (KGs) represent structured collections of facts describing the world in the form of typed relationships between entities Hogan et al. (2020). These collections of facts have been used in a wide range of applications including Web search Eder (2012)), cancer research Saleem et al. (2014), and even entertainment Malyshev et al. (2018). However, most KGs on the Web are far from being complete Nickel et al. (2015). For instance, the birth place of 71% of the persons in Freebase and 66% of the persons in DBpedia is not to be found in the respective KGs. In addition, more than 58% of the scientists in DBpedia are not linked to the predicate that describes what they are known for Krompaß et al. (2015). Identifying such missing links is referred to as *link prediction* Dettmers et al. (2018). Knowledge Graph Embeddings (KGE) approaches map KGs to continuous vector spaces and have been proven to be highly effective and efficient at addressing the task of link prediction Ji et al. (2020); Bordes et al. (2013; 2011); Dettmers et al. (2018).

In this paper, we propose CONEX, a simple but effective new KGE approach. CONEX is a complex-valued convolutional neural model that learns complex-valued vector representations of a given KG by combining a 2D convolution operation with a Hermitian inner product. The motivation behind our approach lies in the following considerations:

1. Convolutional Neural Networks (CNNs) have demonstrated recognition accuracy better than or comparable to humans in several visual recognition tasks, including image recognition, object detection and semantic segmentation Krizhevsky et al. (2012); Girshick et al. (2014); Simonyan & Zisserman (2014). Parallel to the successful application of CNN in computer vision, Dettmers et al. (2018) leverages a multi-layer CNN for learning continuous vector representations of KGs and reaches a state-of-the-art performance in link prediction.
2. Learning complex-valued vector representations of KGEs has been proven to be an effective technique for link prediction Trouillon et al. (2016); Sun et al. (2019b).

We evaluate our approach against 37 state-of-the-art approaches on four benchmark datasets often used in the literature. Overall, our results suggest that CONEX outperforms current state-of-the-art approaches (including RotatE, ConvE, QuatE, ComplEx and TuckER Trouillon et al. (2016); Bordes et al. (2013); Nickel et al. (2011); Shi & Weninger (2017); Balažević et al. (2019b)), in terms of Mean Reciprocal Rank (MRR) and Hits at N (H@N).

Table 1: State-of-the-art KGE models with training strategies. $\mathbf{e}$ denotes embeddings, $\overline{\mathbf{e}} \in \mathbb{C}^d$ corresponds to the complex conjugate of $\mathbf{e}_.$. $*$ denotes a 2D convolution operation with ω kernel. f denotes a non-linear function. The tensor product along the n-th mode is denoted by $\times_n$ and the core tensor is represented by $\mathcal{W}$. $\otimes, \circ, \cdot$ denotes the Hamilton, the Hadamard and an inner product, respectively. $\mathcal{H}$ corresponds to scoring function of ComplEx and $\mathcal{V}$ corresponds to convolution operation followed by linear classifier. $\langle x, y, z \rangle = \sum_i x_i y_i z_i$ denotes the tri-linear dot product. MSE, MR, BCE and CE denotes mean squared error, margin ranking, binary cross entropy and cross entropy.

Model	Scoring Function	VectorSpace	Loss	Training	Optimizer	Regularizer
RESCAL (2011)	$\mathbf{e}_s \cdot \mathcal{W}_p \cdot \mathbf{e}_o$	$\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}$	MSE	Full	ALS	L2
DistMult (2015)	$\langle \mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \rangle$	$\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{R}$	MR	NegSamp	Adagrad	Weighted L2
ComplEx (2016)	$\mathrm{Re}(\langle \mathbf{e}_s, \mathbf{e}_p, \overline{\mathbf{e}_o} \rangle)$	$\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{C}$	BCE	NegSamp	Adagrad	Weighted L2
ConvE (2018)	$f(\mathrm{vec}(f([\mathbf{e}_s; \mathbf{e}_p] * \omega))\mathbf{W})\mathbf{e}_o$	$\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{R}$	BCE	KvsAll	Adam	Dropout,BatchNorm
TuckER (2019b)	$\mathcal{W} \times_1 \mathbf{e}_s \times_2 \mathbf{e}_p \times_3 \mathbf{e}_o$	$\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{R}$	BCE	KvsAll	Adam	Dropout,BatchNorm
RotatE (2019b)	$-\Vert \mathbf{e}_s \circ \mathbf{e}_p - \mathbf{e}_o \Vert$	$\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{C}$	CE	NegSamp	Adam	-
QuatE (2019)	$\mathbf{e}_s \otimes \mathbf{e}_p \cdot \mathbf{e}_o$	$\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{H}$	CE	NegSamp	Adam	L2
ConEx-ours-	$\mathcal{H}(\langle \mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \rangle ; \mathcal{V}(\mathbf{e}_s, \mathbf{e}_p))$	$\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{C}$	BCE	KvsAll	Adam	Dropout,BatchNorm

2 Related work

A wide range of works have investigated KGE to address various tasks such as link prediction, question answering, item recommendation and knowledge graph completion Nickel et al. (2011); Huang et al. (2019). We refer to Nickel et al. (2015); Wang et al. (2017); Cai et al. (2018); Ji et al. (2020); Qin et al. for recent surveys and give a brief overview of selected KGE techniques by using terminology adapted to RDF KGs. Table 1 shows scoring functions of state-of-the-art KGE models.

RESCAL Nickel et al. (2011) is a bilinear model that computes a three-way factorization of a third-order adjacency tensor representing the input KG. RESCAL captures various types of relations in the input KG but is limited in its scalability as it has quadratic complexity in the factorization rank Nickel et al. (2015); Trouillon et al. (2017). DistMult Yang et al. (2015) can be seen as an efficient extension of RESCAL with a diagonal matrix per relations to reduce complexity of RESCAL Balažević et al. (2019b). DistMult performs poorly on antisymmetric relations while performing well on symmetric relations Trouillon et al. (2017). ComplEx Trouillon et al. (2016) extends DistMult by learning representations in a complex vector space. Complex is able to infer both symmetric and antisymmetric relations via a Hermitian inner product of embeddings that involves the conjugate-transpose of one of the two input vectors. ComplEx yields state-of-the-art performance on the link prediction task while leveraging linear space and time complexity of the dot products. Inspired by Euler's identity, RotatE Sun et al. (2019b) employs a rotational model taking predicates as rotations from subjects to objects in complex space via the element-wise Hadamard product Ji et al. (2020). RotatE performs well on composition relations while ComplEx performs poorly Sun et al. (2019b). QuatE Zhang et al. (2019) extends the complex-valued space into hypercomplex by a quaternion with three imaginary components, where the Hamilton product is used as compositional operator for hypercomplex valued-representations. ConvE Dettmers et al. (2018) applies a 2D convolution to model the interactions between entities and relations. Through interactions captured by 2D convolution, ConvE yields a state-of-art performance in link prediction. TuckER Balažević et al. (2019b) performs a Tucker decomposition on the binary tensor representing the input KG the tensor of triples, which enables multi-task learning between different relations via the core tensor.

3 Preliminaries and Notation

In this section, we present the core notation and terminology used throughout this paper.

Knowledge Graphs A KG $\mathcal{G}$ is a set of triples $(s, p, o) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ where $\mathcal{E}$ is the set of all resources (also called *entities*) and P is the set of all properties (also called *relations*).

Link Prediction Predicting missing links refers to predicting the existence of typed directed edges in $\mathcal{G}$. This non-trivial endeavor is known as *link prediction* and is a subtask of Knowledge Graph Completion (KGC) Ji et al. (2020). The link prediction problem is often formalised by learning a scoring function $\phi : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \mapsto \mathbb{R}$ Nickel et al. (2015); Ji et al. (2020) ideally characterized by $\phi((\mathrm{s}, \mathrm{p}, \mathrm{o})) >> \phi((\mathrm{x}, \mathrm{y}, \mathrm{z}))$ if $(\mathrm{s}, \mathrm{p}, \mathrm{o}) \in \mathcal{G} \wedge (\mathrm{x}, \mathrm{y}, \mathrm{z}) \notin \mathcal{G}$.

The Convolution Operation A convolution is an integral expressing the amount of overlap of one function f as it is shifted over another function g Goodfellow et al. (2016). Formally, the convolution operation over a finite range $[0, \tau]$ is given by

$$(f * g)(t) = \int_0^\tau f(\tau) g(t - \tau) d\tau \tag{1}$$

where $*$ denotes the convolution operation. f is often called the input while g is called the kernel (or filter). The output of the $f * g$ is referred as the feature map Goodfellow et al. (2016). In practice, the input often denotes a multidimensional vector of data while the kernel is a multidimensional array of parameters that are adapted by the learning algorithm. Suppose that f represents a 2-dimensional image and g denotes a 2-dimensional kernel. Then, Equation (1) can be rewritten as

$$(f * g)(i, j) \quad = \quad \sum_m \sum_n f(m, n) g(i - m, j - n), \tag{2}$$

where i, j denotes the coordinate in 2-D input. We refer to Goodfellow et al. (2016) for more details on the convolution operation.

4 Approach

In this section, we introduce our KGE approach dubbed ConEx (convolutional complex knowledge graph embeddings). Section 4.1 presents the intuition behind our model. Section 4.2 elucidates the formal workings of ConEx.

4.1 Intuition

The approaches presented in Trouillon et al. (2017); Sun et al. (2019b); Sadeghi et al. (2020) suggest that being able to support symmetric, antisymmetric and inverse relations as well as compositions is a desirable feature of KGEs. Previous works Trouillon et al. (2016); Sun et al. (2019b) show that ComplEx is able to successfully capture symmetric, inverse and antisymmetric relations but fails to capture composite relations because a bijection mapping from subject to object, via relations, is not modelled Sun et al. (2019b). We are interested in the composition of a 2D convolution with a Hermitian inner product. We aim to benefit from the *sparse connectivity*, *parameter sharing* and *equivariant representations* properties of convolutions Goodfellow et al. (2016). This allows us to more accurately capture all four types of relations than approaches which solely apply 2D convolution in $\mathbb{R}$ (e.g., ConvE Dettmers et al. (2018)) or inner products in $\mathbb{C}$ by Hermitian inner products (e.g., ComplEx Trouillon et al. (2016) or the Hadamard product (e.g., RotatE Sun et al. (2019b)).

4.2 ConEx

In this section, we formally elucidate the intuition behind our approach. ConEx is defined as

$$\phi(s, p, o) = \mathcal{H}(\langle \mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \rangle \, ; \mathcal{V}(\mathbf{e}_s, \mathbf{e}_p)) \tag{3}$$

where $\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{C}^d$ and $\mathcal{V}$ denotes a 2D convolution layer with ω kernel over an input of $\mathbb{R}^{4 \times d}$ followed by a linear transformation to project the feature map $\mathcal{T} \in \mathbb{R}^{c \times m \times n}$ into $\mathbb{R}^d$ and c denotes the number of 2D feature maps with dimensions m and n. Formally, $\mathcal{V}$ is defined as

$$\mathcal{V}(\mathbf{e}_s, \mathbf{e}_p) = f\big(\operatorname{vec}(f([\operatorname{Re}(\mathbf{e}_s), \operatorname{Re}(\mathbf{e}_p), \operatorname{Im}(\mathbf{e}_s), \operatorname{Im}(\mathbf{e}_p)] * \omega))\mathbf{W}\big), \tag{4}$$

where f, ω, vec and $\mathbf{W}$ denote rectified linear units, a kernel in a 2D convolution layer, a flatting operation and a projection matrix $\mathbf{W}$, respectively. Re and Im correspond to the real and imaginary parts of a complex number.

$\mathcal{H}$ denotes the composition of $\mathcal{V}$ with a Hermitian inner product of complex-valued vectors and is defined as

$$\begin{aligned} \mathcal{H}(\langle \mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \rangle\, ; x) &= \sum_{k=1}^{d} x_k \mathrm{Re}(\mathbf{e}_s)_k \mathrm{Re}(\mathbf{e}_p)_k \mathrm{Re}(\overline{\mathbf{e}_o})_k \quad (5) \\ &= \langle x, \mathrm{Re}(\mathbf{e}_s), \mathrm{Re}(\mathbf{e}_p), \mathrm{Re}(\mathbf{e}_o)\rangle \\ &\quad + \langle x, \mathrm{Re}(\mathbf{e}_s), \mathrm{Im}(\mathbf{e}_p), \mathrm{Im}(\mathbf{e}_o)\rangle \\ &\quad + \langle x, \mathrm{Im}(\mathbf{e}_s), \mathrm{Re}(\mathbf{e}_p), \mathrm{Im}(\mathbf{e}_o)\rangle \\ &\quad - \langle x, \mathrm{Im}(\mathbf{e}_s), \mathrm{Im}(\mathbf{e}_p), \mathrm{Re}(\mathbf{e}_o)\rangle \quad (6) \end{aligned}$$

where $\overline{\mathbf{e}_o}$ is the conjugate of $\mathbf{e}_o$.

4.2.1 TRAINING.

In the feed-forward pass, we firstly apply a row-vector look-up operation to obtain $\mathbf{e}_s$, $\mathbf{e}_p$ and $\mathbf{e}_o$. Next, we stack the real and imaginary parts of $\mathbf{e}_s$ and $\mathbf{e}_p$, i.e., we compute the vector $[\mathrm{Re}(\mathbf{e}_s), \mathrm{Re}(\mathbf{e}_p), \mathrm{Im}(\mathbf{e}_s), \mathrm{Im}(\mathbf{e}_p)] \in \mathbb{R}^{4\times d}$. Applying a 2D convolution with ω results in obtaining a feature map $\mathcal{T} \in \mathbb{R}^{c\times m\times n}$. Thereafter, we flatten $\mathcal{T}$ with $\mathrm{vec}(.)$ and project it into $\mathbb{R}^d$ by applying the linear transformation $\mathbf{W}$. Next, we compute a Hermitian inner product of $\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o$ as defined in Equation (6). Finally, we compute the score of $(\mathrm{s}, \mathrm{p}, \mathrm{o})$ by applying the logistic sigmoid function $\hat{y} = \sigma(\phi(s, p, o))$ and minimise the following binary cross-entropy loss function:

$$\mathcal{L}(y, \hat{y}) = -\big(y \cdot \log(\hat{y}) + (1-y)\cdot \log(1-\hat{y})\big), \quad (7)$$

where $y = 1$ if $(\mathrm{s}, \mathrm{p}, \mathrm{o}) \in \mathcal{G}$, otherwise $y = 0$.

4.2.2 OPTIMIZATION.

During training, we follow a 1-N scoring regime (with $N = |\mathcal{E}|$) for efficient training Dettmers et al. (2018). In the 1-N scoring regime, a KGE model takes (s, p) as an input and generates $|\mathcal{E}|$ scores for each RDF triple $(\mathrm{s}, \mathrm{p}, \mathrm{x})$ with $x \in \mathcal{E}$. Training with 1-N scoring regime has two advantages: (1) the regime has an effect akin to batch normalization, and (2) faster convergence Dettmers et al. (2018). We also employ the Glorot initialization technique for parameters of CONEX, as using the logistic sigmoid activation often drives the top hidden layer into saturation provided that parameters are randomly initialized Glorot & Bengio (2010). The parameters are optimized using the Adam optimiser Kingma & Ba (2014) in mini-batch fashion. During loss minimization, we employ the early stopping technique Caruana et al. (2001). As a non-linearity in $\mathcal{V}$, we use ReLU for faster training Krizhevsky et al. (2012). To lessen overfitting, we use the dropout technique on each layer Srivastava et al. (2014) and we apply label smoothing Szegedy et al. (2016). Moreover, we perform batch normalisation after each layer Ioffe & Szegedy (2015) to avoid internal covariate shift in the parameters.

5 EXPERIMENTS

We aim to address the following research questions with our experiments:

- Q1: Does CONEX yield competitive results on standard benchmark datasets?
- Q2: How does CONEX compare to the state of the art w.r.t. its complexity in terms of number of parameters?

To address these questions, we relied on the settings described in the following.

5.1 DATASETS

We used the WN18RR Dettmers et al. (2018), FB15K-237 Toutanova et al. (2015), KINSHIP Kok & Domingos (2007) and UMLS Das et al. (2017) benchmark datasets Toutanova et al. (2015); Dettmers et al. (2018). WN18RR is a subset of Wordnet that describes lexical and semantic hierarchies between

Table 2: Number of entities, predicates, and triples in each split for the benchmark datasets. $\mathcal{G}^{\text{Train}}$ and $\mathcal{G}^{\text{Test}}$ indicate the train and test split of the benchmark datasets.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{G}^{\text{Train}}\rvert$	$\lvert\mathcal{G}^{\text{Test}}\rvert$
FB15K-237	14,541	237	272,115	20,466
WN18RR	40,943	11	86,835	3,134
KINSHIP	104	25	8544	1074
UMLS	135	46	5216	661

concepts and involves **symmetric** and **antisymmetric** relation types while FB15K-237 is a subset of Freebase that involves mainly **symmetric**, **antisymmetric** and **composite** relation types Sun et al. (2019b). KINSHIP contains a set of triples that explains kinship relationships among members of the Alyawarra tribe from Central Australia while the Unified Medical Language System (UMLS) dataset is a set of RDF triples describing biomedical concepts such as diseases and relates them to treatments and diagnoses. We followed the recommendation of Dettmers et al. (2018) and did not consider the WN18 and FB15K datasets since Toutanova et al. (2015); Dettmers et al. (2018) show that they suffer from test leakage through inverse relations. An overview of the datasets is provided in Table 2.

5.2 Evaluation Settings

Like in previous works Nguyen (2017); Sun et al. (2019b); Trouillon et al. (2016); Dettmers et al. (2018); Balažević et al. (2019b), we applied the filtered MRR and hits at N (H@N) to evaluate the performance of link prediction approaches. During our evaluation, given (s, p, o), we measure link prediction performances through prediction of missing tail entities. We refer to the supplemental materials for further details pertaining to MRR and H@N.

5.3 Experimental Setup

We selected the hyperparameters of ConEx via a grid search which optimized for MRR on the validation set. The ranges of the hyperparameters for the grid search were defined as follows - d:$\{20, 100, 200\}$, dropout rate:$\{0.1, 0.2, 0.3, 0.4, 0.5\}$, label smoothing :$\{0.0, 0.1\}$ and the number of output channels in the convolution operation c : $\{2, 8, 64\}$. We used early stopping according to MRR as similarly done in Dettmers et al. (2018). After determining the best hyperparameters based on the MRR on the validation dataset, we trained ConEx again with the same hyperparameters on the train set by applying the data augmentation technique from Lacroix et al. (2018); Dettmers et al. (2018) where the reverse triple (o, $\texttt{p}^{-1}$, s) of every (s, p, o) (with p inverse of $\texttt{p}^{-1}$) is added to the validation dataset. We compared the MRR performance on the validation set and decided not to include the data augmentation technique into the grid search as it increased the training time without any significant gains. Throughout our experiments, the seed for the pseudo-random generator was fixed to 1. The best performing parameters for each competing approach can be found at supplemental material. We used the Pytorch Paszke et al. (2017) implementation of DistMult, ComplEx in Balažević et al. (2019a). We used the implementation of TuckER, HypER and ConvE provided in the corresponding papers. Importantly, we used the Glorot initialization technique in parameter initialization, batch normalisation, label smoothing and 1-N scoring regime for DistMult and ComplEx.

6 Results

Table 3 shows the link prediction results on WN18RR and FB15K-237. ConEx outperforms all state-of-the-art approaches w.r.t. H@1 on WN18RR. In particular, ConEx outperforms ComplEx w.r.t. all measures presented in Table 3. This supports our hypothesis, i.e., that the composition of a 2D convolution with a Hermitian inner product allows the better learning of relations in complex spaces. ConEx yields superior performance on FB15K-237 in all metrics. These results are in line with those reported in Sun et al. (2019b): Approaches that are not able to model composite relations perform poorly on FB15K-237 (e.g., DistMult, ComplEx).

Table 3: Link prediction results on WN18RR and FB15K-237. The results presented herein were gathered from the corresponding papers.

	WN18RR				FB15K-237			
	MRR	H@10	H@3	H@1	MRR	H@10	H@3	H@1
CP (Schlichtkrull et al., 2018)	-	-	-	-	.182	.357	.197	.101
HolE (Schlichtkrull et al., 2018)	-	-	-	-	.222	.391	.253	.133
R-GCN+ (Schlichtkrull et al., 2018)	-	-	-	-	.249	.417	.264	.151
R-GCN (Schlichtkrull et al., 2018)	-	-	-	-	.248	.417	-	.151
CP-N3 (Lacroix et al., 2018)	.470	.540	-	-	.360	.540	-	-
CP-FRO (Lacroix et al., 2018)	.46	.48	-	-	.34	.510	-	-
ComplEx-FRO (Lacroix et al., 2018)	.470	.540	-	-	.350	.530	-	-
ComplEx-N3 (Lacroix et al., 2018)	.480	.570	-	-	.370	.560	-	-
Node+LinkFeat (Toutanova & Chen, 2015)	-	-	-	-	.226	.347	-	-
KBLRN (Jiang et al., 2019)	-	-	-	-	.309	.493	-	.219
VR-GCN (Ye et al., 2019)	-	-	-	-	.248	.432	.272	.159
MDE (Sadeghi et al., 2020)	.457	.536	-	-	.288	.484	-	-
TransE (Zhang et al., 2019)	.226	.501	-	-	.294	.465	-	-
ConvKB (Nguyen et al., 2017)	.248	.525	-	-	.329	.517	-	-
KBGAN (Cai & Wang, 2017)	.213	.581	-	-	.278	.458	-	-
TransEdge-CC (Sun et al., 2019a)	.439	.516	-	.411	.310	.482	-	.227
TransEdge-CP (Sun et al., 2019a)	.451	.487	-	.433	.333	.512	-	.243
NKGE (Zhang et al., 2019)	.450	.526	.465	.421	.33	.510	.365	.241
ConvR (Jiang et al., 2019)	.475	.537	.489	.443	.350	.528	.385	.261
TransE-GCN (Cai et al., 2019)	.233	.508	.338	.203	.315	.477	.324	.229
RotatE-GCN (Cai et al., 2019)	.485	.578	.510	.438	.356	.555	.388	.252
ConvE (Dettmers et al., 2018)	.430	.520	.440	.400	.325	.501	.356	.237
OTE (Tang et al., 2019)	.485	**.587**	.502	.437	.351	.537	.388	.258
GC-OTE (Tang et al., 2019)	.491	.583	.511	.442	.361	.550	.267	.237
A2N (Tang et al., 2019)	.450	.510	.460	.420	.317	.486	.348	.232
SACN (Tang et al., 2019)	.470	.540	.480	.430	.352	.536	.385	.261
D4-Gumbel (Xu & Li, 2019)	.486	.557	.505	.442	.300	.496	.332	.204
RotatE (Sun et al., 2019b)	.476	.571	.492	.428	.338	.533	.375	.241
pRotatE (Sun et al., 2019b)	.462	.552	.479	.417	.328	.524	.365	.230
COMPGCN (Vashishth et al., 2019)	.479	.546	.494	.443	.355	.535	.390	.264
QuatE (Zhang et al., 2019)	**.488**	.582	**.508**	.438	.348	.550	.383	.248
DistMult Balažević et al. (2019b)	.430	.490	.440	.390	.241	.419	.263	.155
ComplEx (Balažević et al., 2019b)	.440	.510	.460	.410	.247	.428	.275	.158
TuckER (Balažević et al., 2019b)	.470	.526	.482	.443	.358	.544	.394	.266
HypER (Balažević et al., 2019a)	.465	.522	.477	.436	.341	.520	.376	.252
MuRP (Balazevic et al., 2019)	.481	.566	.495	.440	.335	.518	.367	.243
CONEX (ours)	.472	.523	.484	**.444**	**.393**	**.561**	**.433**	**.306**

Table 4: Link prediction results on KINSHIP and UMLS.

	KINSHIP				UMLS			
	MRR	H@10	H@3	H@1	MRR	H@10	H@3	H@1
ConvE (Dettmers et al., 2018)	.830	.980	.920	.740	.940	.990	.960	.920
GNTP (Minervini et al., 2019)	.719	.958	.815	.719	.841	.986	.941	.732
DistMult	.673	.931	.745	.548	.927	.991	.952	.894
ComplEx	.865	.978	.931	.791	.947	.989	.968	.921
TuckER	.842	**.985**	.907	.758	.907	.997	**.992**	.822
HypER	.676	.910	.754	.554	.755	.933	.839	.652
CONEX (ours)	**.882**	.982	**.944**	**.814**	**.965**	**.994**	.980	**.947**

We compared ConEx against some of the best-performing approaches from Table 3 on two supplementary benchmark datasets. In addition to relying on results presented in previous works Dettmers et al. (2018); Minervini et al. (2019), we optimized and executed DistMult, ComplEx, TuckER and HypER using the optimization settings as for ConEx. Note that these optimizations led to DistMult and ComplEx achieving better results than those provided in the reference literature Trouillon et al. (2017); Das et al. (2017). To the best of our knowledge, TuckER and HypER have not been evaluated on these two datasets to date. Our results are shown in Table 4. ConEx outperforms all other approaches w.r.t. H@1 and MRR. This outcome is well in line with the results from Table 3.

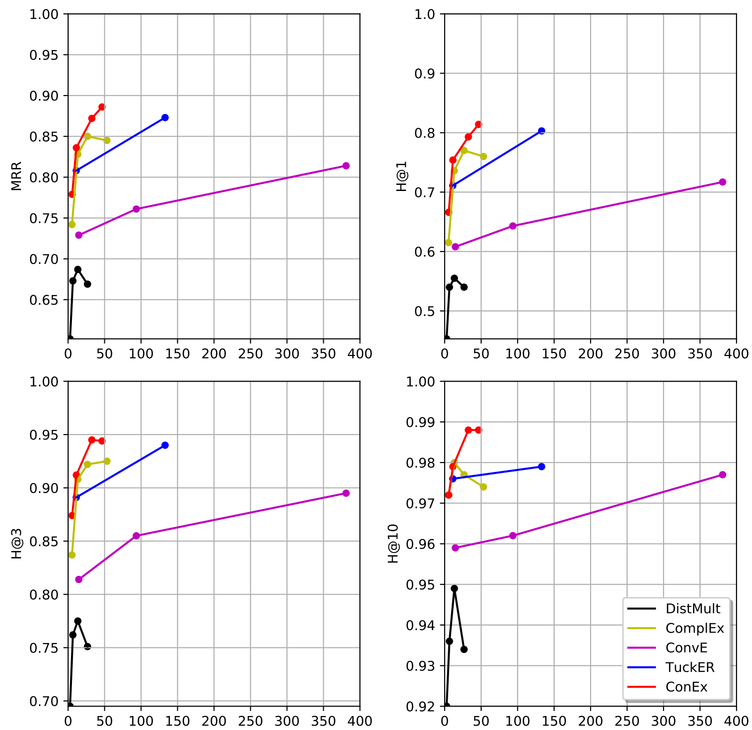

Figure 1: Link prediction results on KINSHIP with different embedding sizes. The x axis denotes the parameters in thousand. The y axis denotes the link prediction performances measured by the respective means.

An important outcome of the aforementioned optimization carried out for Table 4 lies in the comparability of the parameter space of DistMult, ComplEx, ConvE, TuckER and ConEx. We hence studied the link prediction results achieved by these approaches with different embedding sizes. Figure 1 shows that ConEx yields better MRR and H@N performances than DistMult, ComplEx, ConvE, TuckER in various sizes of embeddings $\{20, 50, 100, 200\}$ for the same number of parameters. DistMult scales linearly in terms of its number of parameters as it employs dot products of real-valued embeddings. However, our results suggest that DistMult requires large embedding sizes to achieve its best performance on antisymmetric relations. This entails an explosion of the number of parameters Nickel et al. (2015); Ji et al. (2020); Trouillon et al. (2017), leading to extensive hyperparameter optimization to alleviate possible overfitting. Still, DistMult requires fewer parameters overall than ComplEx and ConEx (assuming a fixed embedding size). In our experiments, ComplEx requires exactly twice the number of parameters used by DistMult due to the real and imaginary parts of the embeddings it generates. The number of parameters of ConEx depends on the number of output channels it uses as well as on the embedding size. Overall, our results suggest that our approach strikes a good middle ground by requiring less parameters than most state-of-the-art approaches while outperforming the state of the art in most experiments.

Table 5: The number of free parameters are obtained from corresponding papers. M denotes million.

	WN18RR	FB15K-237
RotatE (Sun et al., 2019b)	40.95M	29.32M
QuatE (Zhang et al., 2019)	16.38M	5.82M
TuckER (Balažević et al., 2019b)	9.39M	11.00M
CONEX	8.81M	0.64M

Table 6: Link prediction results per relations for CONEX tested on WN18RR. # denotes the number of occurrences of a relation.

Relations	#Train	#Test	MRR	H@10	H@3	H@1
_hypernym	34,796	1,251	.176	.237	.191	.144
_derivationally_related_form	29,715	1,074	.956	.965	.959	.949
_has_part	4,816	172	.097	.203	.087	.058
_synset_domain_topic_of	3,116	114	.510	.623	.544	.438
_instance_hypernym	2,921	122	.615	.803	.697	.508
_also_see	1,299	56	.559	.643	.625	.50
_verb_group	1,138	39	.910	.974	.974	.846
_member_meronym	7,402	253	.040	.087	.036	.016
_member_of_domain_region	923	26	.023	.038	.038	0.0
_member_of_domain_usage	629	24	.007	0.0	0.0	0.0
_similar_to	80	3	1.0	1.0	1.0	1.0

7 DISCUSSION

The superior performance of CONEX stems from the composition of a 2D convolution with a Hermitian inner product of complex-valued embeddings. Applying 2D convolution on complex-valued embeddings of subjects and predicates permits CONEX to recognize interactions between subjects and predicates in the form of complex-valued feature maps. Through the projection of feature maps and their inclusion into a Hermitian inner product involving the conjugate-transpose of complex-valued embeddings of objects, CONEX can accurately infer various types of relations. For instance, CONEX is able to model composition patterns without defining the bijection mapping explicitly. This ability is suggested by Table 3 since WN18RR and FB15K-237 involve antisymmetric and composite relations Sun et al. (2019b). Moreover, Table 5 shows that CONEX requires significantly fewer parameters than RotatE, QuatE and TuckER than WN18RR and FB15K-237. Table 6 and two more tables in the supplemental material explicitly show that CONEX is able to capture various types of relations on benchmark datasets. However, CONEX inaccurately ranks entities with `_member_of_domain_region` and `_member_of_domain_usage`. This may indicate that CONEX is not able model triples where subjects and objects are *loosely* semantically related Allen et al. (2019). Overall, CONEX is more expressive than approaches that solely apply 2D convolution in $\mathbb{R}$ (e.g. ConvE Dettmers et al. (2018)) and solely apply inner products in $\mathbb{C}$ by Hermitian Inner Products (e.g. ComplEx Trouillon et al. (2016) or Hadamard product (e.g. RotatE Sun et al. (2019b)).

8 CONCLUSION AND FUTURE WORK

In this work, we introduce a new approach (called CONEX) for addressing the link prediction problem by learning continuous vector representations for knowledge graphs. CONEX accurately infers the various types of relations by leveraging a composition of a 2D convolution with a Hermitian inner product of complex-valued embeddings. CONEX achieves state-of-the-art performances on standard link prediction datasets while requiring fewer parameters than several state-of-the-art approaches—including QuatE Zhang et al. (2019), RotatE Sun et al. (2019b) and TuckER Balažević et al. (2019b). In future work, we plan to explore combining 2D convolution with Hamilton's Quaternions Zhang et al. (2019).

REFERENCES

Carl Allen, Ivana Balazevic, and Timothy M Hospedales. On understanding knowledge graph representation. *arXiv preprint arXiv:1909.11611*, 2019.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. Multi-relational poincaré graph embeddings. In *Advances in Neural Information Processing Systems*, pp. 4465–4475, 2019.

Ivana Balažević, Carl Allen, and Timothy M Hospedales. Hypernetwork knowledge graph embeddings. In *International Conference on Artificial Neural Networks*, pp. 553–565. Springer, 2019a.

Ivana Balažević, Carl Allen, and Timothy M Hospedales. Tucker: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590*, 2019b.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

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*, pp. 2787–2795, 2013.

Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.

Ling Cai, Bo Yan, Gengchen Mai, Krzysztof Janowicz, and Rui Zhu. Transgcn: Coupling transformation assumptions with graph convolutional networks for link prediction. In *Proceedings of the 10th International Conference on Knowledge Capture*, pp. 131–138, 2019.

Liwei Cai and William Yang Wang. Kbgan: Adversarial learning for knowledge graph embeddings. *arXiv preprint arXiv:1711.04071*, 2017.

Rich Caruana, Steve Lawrence, and C Lee Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*, pp. 402–408, 2001.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. *arXiv preprint arXiv:1711.05851*, 2017.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Jeffrey Scott Eder. Knowledge graph based search system, June 21 2012. US Patent App. 13/404,109.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, et al. Knowledge graphs. *arXiv preprint arXiv:2003.02320*, 2020.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 105–113, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*, 2020.

Xiaotian Jiang, Quan Wang, and Bin Wang. Adaptive convolution for multi-relational learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 978–987, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Stanley Kok and Pedro Domingos. Statistical predicate invention. In *Proceedings of the 24th international conference on Machine learning*, pp. 433–440, 2007.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Denis Krompaß, Stephan Baier, and Volker Tresp. Type-constrained representation learning in knowledge graphs. In *International semantic web conference*, pp. 640–655. Springer, 2015.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. *arXiv preprint arXiv:1806.07297*, 2018.

Stanislav Malyshev, Markus Krötzsch, Larry González, Julius Gonsior, and Adrian Bielefeldt. Getting the most out of wikidata: semantic technology usage in wikipedia's knowledge graph. In *International Semantic Web Conference*, pp. 376–394. Springer, 2018.

Pasquale Minervini, Matko Bošnjak, Tim Rocktäschel, Sebastian Riedel, and Edward Grefenstette. Differentiable reasoning on large knowledge bases and natural language. *arXiv preprint arXiv:1912.10824*, 2019.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. A novel embedding model for knowledge base completion based on convolutional neural network. *arXiv preprint arXiv:1712.02121*, 2017.

Dat Quoc Nguyen. An overview of embedding models of entities and relationships for knowledge base completion. *arXiv preprint arXiv:1703.08098*, 2017.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, volume 11, pp. 809–816, 2011.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Chuan Qin, Hengshu Zhu, Fuzhen Zhuang, Qingyu Guo, Qi Zhang, Le Zhang, Chao Wang, Enhong Chen, and Hui Xiong. A survey on knowledge graph based recommender systems. *SCIENTIA SINICA Informationis*.

Afshin Sadeghi, Damien Graux, H Shariat Yazdi, and Jens Lehmann. Mde: multiple distance embeddings for link prediction in knowledge graphs. ECAI, 2020.

Muhammad Saleem, Maulik R Kamdar, Aftab Iqbal, Shanmukha Sampath, Helena F Deus, and Axel-Cyrille Ngonga Ngomo. Big linked cancer data: Integrating linked tcga and pubmed. *Journal of web semantics*, 27:34–41, 2014.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pp. 593–607. Springer, 2018.

Baoxu Shi and Tim Weninger. ProjE: Embedding projection for knowledge graph completion. In *AAAI*, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 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.

Zequn Sun, Jiacheng Huang, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. Transedge: Translating relation-contextualized embeddings for knowledge graphs. In *International Semantic Web Conference*, pp. 612–629. Springer, 2019a.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019b.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Yun Tang, Jing Huang, Guangtao Wang, Xiaodong He, and Bowen Zhou. Orthogonal relation transforms with graph context modeling for knowledge graph embedding. *arXiv preprint arXiv:1911.04910*, 2019.

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*, pp. 57–66, 2015.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1499–1509, 2015.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pp. 2071–2080, 2016.

Théo Trouillon, Christopher R Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. Knowledge graph completion via complex tensor factorization. *The Journal of Machine Learning Research*, 18(1):4735–4772, 2017.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*, 2019.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12): 2724–2743, 2017.

Canran Xu and Ruijiang Li. Relation embedding with dihedral group in knowledge graph. *arXiv preprint arXiv:1906.00687*, 2019.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*, 2015.

Rui Ye, Xin Li, Yujie Fang, Hongyu Zang, and Mingzhong Wang. A vectorized relational graph convolutional network for multi-relational network alignment. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 4135–4141, 2019.

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