Building Bridges: Knowledge Graph Embeddings Respecting Logical Rules

Aleksandar Pavlović[1], Emanuel Sallinger[1,2]

[1]TU Wien, Austria
[2]University of Oxford, United Kingdom

Abstract
Knowledge graphs (KGs) are typically highly incomplete. Therefore substantial research has been directed toward typically machine-learning-based approaches for knowledge graph completion (KGC), i.e., predicting missing triples from the data stored in the KG. KG embedding models (KGEs) have yielded promising results for KGC. In practice, the data management community typically represents major properties of data through constraints, axioms, or dependencies expressed as logical rules. However, any current KGE cannot capture vital logical rules, i.e., infer missing triples while adhering to such rules. For instance, correctly capturing general composition and jointly capturing composition and hierarchy rules is still an open problem. This work introduces the ExpressivE model that bridges this gap between the data management and machine learning community. ExpressivE embeds pairs of entities as points and relations as hyper-parallelograms in the virtual triple space $\mathbb{R}^{2d}$. This model design allows ExpressivE to capture a rich set of logical rules jointly and display any supported rule through the spatial relation of hyper-parallelograms, additionally offering an intuitive and consistent geometric interpretation of ExpressivE embeddings and captured rules. Experimental results on standard KGC benchmarks reveal that ExpressivE is competitive with state-of-the-art KGEs and even significantly outperforms them on WN18RR. This short paper is based on our recently published ICLR 2023 paper [1].

Keywords
Data Management, Logical Rules, Knowledge Graph Embedding, Knowledge Graph Completion

1. Introduction

One of the key challenges in the data management community is to bring together *machine learning* models and – typically logic-based – *data management* approaches. This challenge is especially apparent in the field of *graph data management*, particularly when considering knowledge graphs (KGs) that are typically highly incomplete [2]: On the one hand, the use of machine learning models, called *knowledge graph embedding models* (KGEs), has achieved promising results for *knowledge graph completion* (KGC) [3], i.e., for automatically predicting missing triples. On the other hand, the data management community typically represents major properties of data through *constraints*, *axioms*, or *dependencies* expressed as *logical rules*.

However, there is a major challenge in this: Many KGEs cannot respect vital logical rules – termed *capturing* rules – which describes a KGE's ability to infer missing triples while adhering to such logical rules. *Composition* of relations is a particularly important constraint or

AMW 2023: 15th Alberto Mendelzon International Workshop on Foundations of Data Management
aleksandar.pavlovic@tuwien.ac.at (A. Pavlović); emanuel.sallinger@tuwien.ac.at (E. Sallinger)
0000-0001-6887-9515 (A. Pavlović); 0000-0001-7441-129X (E. Sallinger)

dependency for data management – even more so in graph data management, where it allows to describe paths. Recently, however, it was discovered that existing KGEs only capture a fairly limited notion of composition [4, 5, 6, 7], solely capturing *compositional definition*, not *general composition* (see Table 1 for the defining formulas). Even more, while existing KGEs capture hierarchy [8, 9, 10, 5] and compositional definition [11, 12, 4, 6] individually, they cannot capture both rules simultaneously (see Table 1).

Table 1
This table lists logical rules that several KGEs can capture, where ✓ represents that the rule is supported and ✗ that it is not supported.

Logical Rule	ExpressivE	BoxE	RotatE	TransE	DistMult	ComplEx
Symmetry: $r_1(X,Y) \Rightarrow r_1(Y,X)$	✓	✓	✓	✗	✓	✓
Anti-symmetry: $r_1(X,Y) \Rightarrow \neg r_1(Y,X)$	✓	✓	✓	✓	✗	✓
Inversion: $r_1(X,Y) \Leftrightarrow r_2(Y,X)$	✓	✓	✓	✓	✗	✓
Comp. def.: $r_1(X,Y) \wedge r_2(Y,Z) \Leftrightarrow r_3(X,Z)$	✓	✗	✓	✓	✗	✗
Gen. comp.: $r_1(X,Y) \wedge r_2(Y,Z) \Rightarrow r_3(X,Z)$	✓	✗	✗	✗	✗	✗
Hierarchy: $r_1(X,Y) \Rightarrow r_2(X,Y)$	✓	✓	✗	✗	✓	✓
Intersection: $r_1(X,Y) \wedge r_2(X,Y) \Rightarrow r_3(X,Y)$	✓	✓	✓	✓	✗	✗
Mutual exclusion: $r_1(X,Y) \wedge r_2(X,Y) \Rightarrow \bot$	✓	✓	✓	✓	✓	✓

While the extensive research on composition [11, 12, 4, 6] and hierarchy [8, 10, 9, 5] highlights their importance, any KGE so far is incapable of: (1) capturing general composition, (2) capturing composition and hierarchy jointly, and (3) providing an intuitive geometric interpretation of captured rules. Thus, in this paper, our goal is to overcome these limitations by introducing a KGE that captures a wide range of logical rules relevant to the data management community:

- We introduce the **ExpressivE** model and the *virtual triple space* it is based on, which allows for an intuitive geometric interpretation of captured rules.
- We prove that our model captures **any rule in Table 1**, the first such KGE.
- We evaluate ExpressivE on KGC, finding that it is competitive with state-of-the-art (SOTA) KGEs, even significantly outperforming them on some datasets.

2. Preliminaries

KGs can be represented as large collections of triples $r_i(e_h, e_t)$ over a finite set of relations $r_i \in \mathbf{R}$ and entities $e_h, e_t \in \mathbf{E}$, where we call e_h the triple's *head* and e_t its *tail*. In what follows, we assume the standard definition of capturing logical rules [12, 5, 1]. This means intuitively that a KGE captures a rule if a set of parameters exists such that the logical rule is captured *exactly* (i.e., any logically inferrable triple is predicted by the KGE) and *exclusively* (i.e., no unwanted rule is supported by the KGE's predictions).

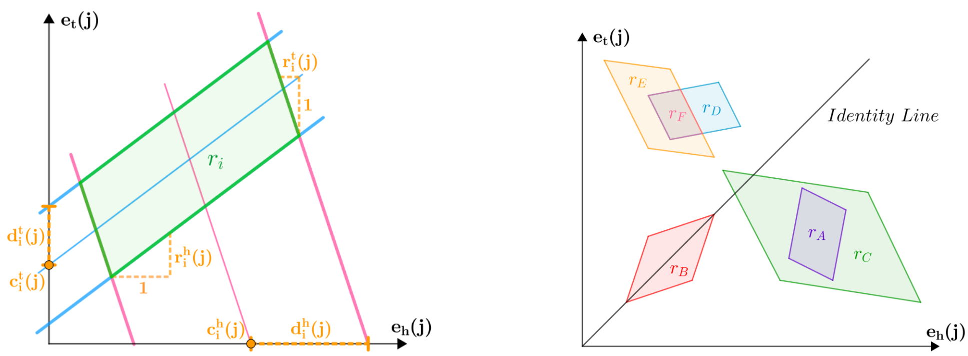
3. ExpressivE and the Virtual Triple Space

ExpressivE embeds entities $e_j \in \boldsymbol{E}$ via a vector $\boldsymbol{e_j} \in \mathbb{R}^d$, representing points in the embedding space $\mathbb{R}^d$ and relations $r_i \in \boldsymbol{R}$ as hyper-parallelograms in the virtual triple space $\mathbb{R}^{2d}$ (see Figure 1a for a visual representation). More specifically, ExpressivE assigns to a relation r_i for each of its arity positions $p \in \{h, t\}$: (1) a *slope vector* $\boldsymbol{r_i^p} \in \mathbb{R}^d$, (2) a *center vector* $\boldsymbol{c_i^p} \in \mathbb{R}^d$, and (3) a *width vector* $\boldsymbol{d_i^p} \in (\mathbb{R}_{\geq 0})^d$. Intuitively, these vectors define the slope $\boldsymbol{r_i^p}$ of the hyper-parallelogram's boundaries, its center $\boldsymbol{c_i^p}$, and width $\boldsymbol{d_i^p}$. A triple $r_i(e_h, e_t)$ is captured to be true in an ExpressivE model if its relation and entity embeddings satisfy the following inequalities:

$$(\boldsymbol{e_h} - \boldsymbol{c_i^h} - \boldsymbol{r_i^t} \odot \boldsymbol{e_t})^{|.|} \preceq \boldsymbol{d_i^h} \tag{1}$$

$$(\boldsymbol{e_t} - \boldsymbol{c_i^t} - \boldsymbol{r_i^h} \odot \boldsymbol{e_h})^{|.|} \preceq \boldsymbol{d_i^t} \tag{2}$$

Where $\boldsymbol{x}^{|.|}$ represents the element-wise absolute value of a vector $\boldsymbol{x}$, $\odot$ represents the Hadamard product, and $\preceq$ represents the element-wise less or equal operator. As it is very complex to interpret this model in the embedding space $\mathbb{R}^d$, we introduce the *virtual triple space* $\mathbb{R}^{2d}$ next that eases reasoning about ExpressivE's parameters and inference capabilities.



(a) (b)

Figure 1: (a) Interpretation of relation parameters (orange dashed) as a parallelogram (green solid) in the j-th correlation subspace; (b) Multiple relation embeddings with the following properties: Symmetry (r_B), Anti-Symmetry (r_A, r_D, r_E, r_F), Inversion ($r_D = r_A^{-1}$), Hierarchy $r_A(X,Y) \Rightarrow r_C(X,Y)$, Intersection $r_D(X,Y) \wedge r_E(X,Y) \Rightarrow r_F(X,Y)$, Mutual Exclusion (e.g., $r_A(X,Y) \wedge r_B(X,Y) \Rightarrow \bot$).

Interpretation. We construct the virtual triple space $\mathbb{R}^{2d}$ by concatenating the head $\boldsymbol{e_h}$ and tail embeddings $\boldsymbol{e_t}$. We call the 2-dimensional sub-space of $\mathbb{R}^{2d}$, created from the j-th dimension of $\boldsymbol{e_h}$ and $\boldsymbol{e_t}$, the *j-th correlation subspace*, as it visualizes the captured logical rules of this dimension. As visualized in Figure 1a, the relation parameters define a hyper-parallelogram in $\mathbb{R}^{2d}$. Using these notions, we analyze ExpressivE's theoretical capabilities next.

4. Theoretical and Empirical Results

Expressiveness. It is vital for a KGE to be *fully expressive* [5], i.e., to be able to represent any graph G over $\boldsymbol{R}$ and $\boldsymbol{E}$, as otherwise, the KGE may underfit certain KGs severely. Theorem 4.1 proves that ExpressivE is fully expressive.

Theorem 4.1 (Expressive Power). *ExpressivE can capture any arbitrary graph G over $\boldsymbol{R}$ and $\boldsymbol{E}$ if the embedding dimensionality d is at least in $O(|\boldsymbol{E}| * |\boldsymbol{R}|)$.*

Proof Sketch. Theorem 4.1 is proven by induction, starting with an ExpressivE embedding that captures the complete graph, i.e., any triple over $\boldsymbol{E}$ and $\boldsymbol{R}$ is true. Each induction step shows that we can alter the ExpressivE embedding to make an arbitrarily picked triple of the form $r_i(e_j, e_k)$ with $r_i \in \boldsymbol{R}$, $e_j, e_k \in \boldsymbol{E}$ and $e_j \neq e_k$ false. Finally, we add $|\boldsymbol{E}| * |\boldsymbol{R}|$ dimensions to make any self-loop – i.e., any triple of the form $r_i(e_j, e_j)$ with $r_i \in \boldsymbol{R}$ and $e_j \in \boldsymbol{E}$ – false. □

Logical Rules. Theorem 4.2 reveals that ExpressivE can capture any of the most prominently analyzed rules [11, 12, 8, 10, 9, 5] listed in Table 1.

Theorem 4.2. *ExpressivE captures (a) symmetry, (b) anti-symmetry, (c) inversion, (d) hierarchy, (e) intersection, (f) mutual exclusion, (g) general composition, and (h) compositional definition.*

Figure 1b shows how several one-dimensional ExpressivE embeddings capture rules (a)-(f). ExpressivE captures: (a) *symmetry* via symmetric hyper-parallelograms, (b) *anti-symmetry* via hyper-parallelograms that do not overlap with their mirror image, (c) *inversion* via r_2's hyper-parallelogram being the mirror image of r_1's, (d) *hierarchy* via r_2's hyper-parallelogram subsuming r_1's, (e) *intersection* via r_3's hyper-parallelogram subsuming the intersection of r_1's and r_2's, and (f) *mutual exclusion* via non-overlapping hyper-parallelograms.

Composition. Capturing rules (g)-(h) is more complex. In particular, compositional definition is of the form $r_1(X,Y) \wedge r_2(Y,Z) \Leftrightarrow r_d(X,Z)$, where we call r_d the *compositionally defined relation.* This rule defines a relation r_d that describes the start and end entities of a path $X \xrightarrow{r_1} Y \xrightarrow{r_2} Z$. Since any two r_1 and r_2 can instantiate the body of a compositional definition rule, any such pair may produce a new r_d. Interestingly, compositional definition translates analogously into the virtual triple space: Intuitively, this means that the embeddings of any two r_1 and r_2 define for r_d a *convex* region — which we call the *compositionally defined region* — that captures $r_1(X,Y) \wedge r_2(Y,Z) \Leftrightarrow r_d(X,Z)$, leading to Theorem 4.3. Based on this insight, ExpressivE captures compositional definition by embedding r_d with the compositionally defined region, defined by the embeddings of r_1 and r_2. Furthermore, ExpressivE captures general composition by embedding r_d with a hyper-parallelogram that subsumes the compositionally defined region. Finally, capturing composition through the subsumption of spatial regions allows ExpressivE to provably capture composition for *1-N*, *N-1*, and *N-M* relations.

Theorem 4.3. *Let $r_1, r_2, r_d \in \boldsymbol{R}$ be relations, $\boldsymbol{s_1}, \boldsymbol{s_2}$ be their ExpressivE embeddings, and assume $r_1(X,Y) \wedge r_2(Y,Z) \Leftrightarrow r_d(X,Z)$ holds. Then there exists a region $\boldsymbol{s_d}$ in the virtual triple space $\mathbb{R}^{2d}$ such that (i) $\boldsymbol{s_1}, \boldsymbol{s_2}$, and $\boldsymbol{s_d}$ capture $r_1(X,Y) \wedge r_2(Y,Z) \Leftrightarrow r_d(X,Z)$ and (ii) $\boldsymbol{s_d}$ is* convex.

KGE Families. Interpretable KGEs consist of three families [1]: *Functional KGEs*, embedding relations as functions; *bilinear KGEs*, embedding relations as bilinear products; and *spatial KGEs*, embedding relations as regions. ExpressivE is the first KGE belonging to both the spatial and functional family. BoxE [5] is its closest spatial, and RotatE [12] is its closest functional relative.

Space Complexity. For a d-dimensional embedding, RotatE and BoxE have each $(2|\boldsymbol{E}| + 2|\boldsymbol{R}|)d$, whereas ExpressivE has $(|\boldsymbol{E}| + 6|\boldsymbol{R}|)d$ parameters, where $|\boldsymbol{E}|$ is the number of entities and $|\boldsymbol{R}|$ the number of relations. Since $|\boldsymbol{R}| << |\boldsymbol{E}|$ in most graphs, ExpressivE almost *halves* the number of parameters for a d-dimensional embedding compared to BoxE and RotatE.

Table 2
KGC performance of ExpressivE and SOTA KGEs.

Family	Model	WN18RR				FB15k-237			
		H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR
Func. / Spat.	ExpressivE	**.464**	**.522**	.597	**.508**	**.256**	**.387**	.535	**.350**
	BoxE [5]	.400	.472	.541	.451	.238	.374	**.538**	.337
	RotatE [12]	.428	.492	.571	.476	.241	.375	.533	.338
	TransE [12]	.013	.401	.529	.223	.233	.372	.531	.332
Bilinear	DistMult [13, 14]	-	-	.531	.452	-	-	.531	.343
	ComplEx [13, 14]	-	-	**.547**	**.475**	-	-	.536	.348
	TuckER [15]	**.443**	**.482**	.526	.470	**.266**	**.394**	**.544**	**.358**

Benchmark Results. Table 2 reveals that ExpressivE, with only *half* the number of parameters of BoxE and RotatE, performs best among its own model family on FB15k-237 and is competitive with TuckER, especially in MRR. Even more, ExpressivE *outperforms all* competing KGEs significantly on WN18RR. The significant performance increase of ExpressivE on WN18RR is likely due to WN18RR containing both hierarchy and composition rules in contrast to FB15k-237 (similar to the discussion of [5]). Thus, ExpressivE is highly parameter efficient compared to related KGEs while reaching competitive performance on FB15k-237 and even new SOTA performance on WN18RR, supporting the extensive theoretical results of our paper.

5. Conclusion

In this work, we have introduced ExpressivE, a KGE that (1) captures a wide range of logical rules relevant to data management (including general composition and hierarchy), (2) provides an intuitive geometric interpretation of captured rules, and (3) brings together the ability to capture important types of rules with SOTA KGC performance. To facilitate reproducibility and reusability, we provide ExpressivE's code in a public GitHub repository[1].

Acknowledgments

This work has been funded by the Vienna Science and Technology Fund (WWTF) [10.47379/VRG18013, 10.47379/NXT22018, 10.47379/ICT2201]; and the Christian Doppler Research Association (CDG) JRC LIVE.

References

[1] A. Pavlovic, E. Sallinger, Expressive: A spatio-functional embedding for knowledge graph completion, in: The Eleventh International Conference on Learning Representations, 2023. URL: https://openreview.net/forum?id=xkev3_np08z.

[1]https://github.com/AleksVap/ExpressivE

[2] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, D. Lin, Knowledge base completion via search-based question answering, in: Proceedings of the 23rd International Conference on World Wide Web, WWW '14, Association for Computing Machinery, 2014, p. 515–526.

[3] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE Transactions on Knowledge and Data Engineering (2017) 2724–2743.

[4] S. Zhang, Y. Tay, L. Yao, Q. Liu, Quaternion knowledge graph embeddings, in: H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32: Conference on Neural Information Processing Systems, 2019, pp. 2731–2741.

[5] R. Abboud, İ. İ. Ceylan, T. Lukasiewicz, T. Salvatori, Boxe: A box embedding model for knowledge base completion, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Conference on Neural Information Processing Systems, 2020.

[6] H. Lu, H. Hu, Dense: An enhanced non-abelian group representation for knowledge graph embedding, CoRR abs/2008.04548 (2020). arXiv:2008.04548.

[7] C. Gao, C. Sun, L. Shan, L. Lin, M. Wang, Rotate3d: Representing relations as rotations in three-dimensional space for knowledge graph embedding, in: M. d'Aquin, S. Dietze, C. Hauff, E. Curry, P. Cudré-Mauroux (Eds.), CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, ACM, 2020, pp. 385–394.

[8] B. Yang, W. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, 2015.

[9] S. M. Kazemi, D. Poole, Simple embedding for link prediction in knowledge graphs, in: S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems 31: Conference on Neural Information Processing Systems, 2018, pp. 4289–4300.

[10] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: M. Balcan, K. Q. Weinberger (Eds.), Proceedings of the 33nd International Conference on Machine Learning, volume 48 of *JMLR Workshop and Conference Proceedings*, JMLR.org, 2016, pp. 2071–2080.

[11] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: C. J. C. Burges, L. Bottou, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26: 27th Conference on Neural Information Processing Systems 2013., 2013, pp. 2787–2795.

[12] Z. Sun, Z. Deng, J. Nie, J. Tang, Rotate: Knowledge graph embedding by relational rotation in complex space, in: 7th International Conference on Learning Representations, 2019.

[13] D. Ruffinelli, S. Broscheit, R. Gemulla, You CAN teach an old dog new tricks! on training knowledge graph embeddings, in: 8th International Conference on Learning Representations, 2020.

[14] B. Yang, W. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: Proceedings of the Third International Conference on Learning Representations, 2015.

[15] I. Balazevic, C. Allen, T. M. Hospedales, Tucker: Tensor factorization for knowledge graph completion, CoRR abs/1901.09590 (2019). arXiv:1901.09590.