

Raising the Efficiency of Knowledge Graph Embeddings While Respecting Logical Rules

Aleksandar Pavlović¹, Emanuel Sallinger^{1,2}

¹TU Wien, Austria

²University of Oxford, United Kingdom

Abstract

Knowledge graphs (KGs) are highly incomplete. As a result, researchers have proposed mostly machine-learning-based methods for knowledge graph completion (KGC), which is the task of predicting missing links from the information kept in the KG. Geometric KG embedding models (gKGEs) have demonstrated strong KGC results while providing the ability to respect major characteristics of KGs, typically represented in the form of logical rules by the data management community. However, for strong KGC performance, most gKGEs require *high embedding dimensionalities* or *complex embedding spaces*, severely restricting their time and space efficiency. This work addresses these challenges by proposing SpeedE, a lightweight Euclidean gKGE that (1) respects a set of core logical rules relevant to the data management community; (2) outperforms state-of-the-art gKGEs, particularly on YAGO3-10 and WN18RR; and (3) greatly boosts their efficiency, in particular requiring only a quarter of the parameters and a fifth of the training time of the state-of-the-art ExpressivE model on WN18RR to achieve competitive KGC performance. This extended abstract is based on our recently published NAACL 2024 paper [1].

Keywords

Efficiency, Scalability, Data Management, Logical Rules, Knowledge Graph Completion

1. Introduction

Geometric knowledge graph embedding models (gKGEs) represent entities and relations of a *knowledge graph* (KG) as geometric shapes in the semantic vector space. gKGEs achieved promising performance on *knowledge graph completion* (KGC) and knowledge-driven applications [2, 3]; while allowing for an intuitive *geometric interpretation* of their captured patterns [4, 5, 6]. Recently, gKGEs with increasingly *more complex* embedding spaces were explored [7, 8, 9]. However, more complex embedding spaces typically require more costly operations or more parameters, lowering their time and space efficiency compared to Euclidean gKGEs [10]. Even more, most gKGEs require *high-dimensional embeddings* to reach good KGC performance, increasing their time and space requirements [11, 10]. Thus, the need for (1) complex embedding spaces and (2) high-dimensional embeddings lowers the efficiency of gKGEs, hindering their application in resource-constrained environments, especially in mobile smart devices [7, 8, 10].

Challenge and Methodology. Although there has been much work on scalable gKGEs, any such work has focused exclusively on either reducing the embedding dimensionality [12, 11, 13]

AMW 2024: 16th Alberto Mendelzon International Workshop on Foundations of Data Management, September 30th–October 4th, 2024, Mexico City, Mexico

✉ 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)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



or using simpler embedding spaces [14, 15, 6], thus addressing only one side of the efficiency problem. Facing these challenges, this work aims to design a *Euclidean* gKGE that performs well on KGC under *low-dimensional* conditions, reducing its storage requirements, inference, and training time. To reach this goal, we analyze ExpressivE [6], an Euclidean gKGE that has shown promising performance on KGC under high-dimensional conditions.

Contribution. Based on ExpressivE, we propose the lightweight SpeedE model that (1) halves ExpressivE’s inference time and (2) significantly improves its KGC performance. We evaluate SpeedE on the three standard KGC benchmarks, WN18RR, FB15k-237, and YAGO3-10, finding that it (3) is competitive with SotA gKGEs on FB15k-237 and even outperforms them significantly on WN18RR and YAGO3-10. Moreover, we find that (4) on WN18RR SpeedE requires solely a fourth of ExpressivE’s number of parameters and solely a fifth of its training time to reach the same KGC performance (see Table 2 in Section 4).

2. Preliminaries

KGs can be viewed as sets of triples $r_i(e_h, e_t)$ over a finite set of relations $r_i \in \mathbf{R}$ and entities $e_h, e_t \in \mathbf{E}$. Given a triple $r_i(e_h, e_t)$, e_h is called its *head* and e_t its *tail*. Henceforth, we use the standard definition of capturing rules [7, 16, 6], which intuitively states that a KGE captures a rule if there is a parameter set such that the KGE captures the rule *exactly* (i.e., it predicts any logically inferrable triple) and *exclusively* (i.e., it does not capture any undesired rule).

3. Min_SpeedE and SpeedE

Min_SpeedE and SpeedE are Euclidean gKGEs based on ExpressivE [6]. Similarly to Pavlović and Sallinger [6], Min_SpeedE embeds entities $e_h \in \mathbf{E}$ via vectors $\mathbf{e}_h \in \mathbb{R}^d$ and relations $r_j \in \mathbf{R}$ via hyper-parallellograms in \mathbb{R}^{2d} . In contrast to ExpressivE, which parameterizes a hyper-parallellogram of a relation r_j with three vectors, Min_SpeedE solely uses a scalar width parameter w and two vectors: a *slope vector* $\mathbf{s}_j \in \mathbb{R}^{2d}$ representing the slopes of its boundaries and a *center vector* $\mathbf{c}_j \in \mathbb{R}^{2d}$ representing its center. The main difference between Min_SpeedE and ExpressivE is that Min_SpeedE uses a constant width parameter w , thereby, halving ExpressivE’s inference time, as we shall see soon. At an intuitive level, a triple $r_j(e_h, e_t)$ is captured to be *true* by a Min_SpeedE embedding if the concatenation of its head and tail embeddings is within r_j ’s hyper-parallellogram. Formally, this means that a triple $r_j(e_h, e_t)$ is true if the following is satisfied:

$$(\mathbf{e}_{ht} - \mathbf{c}_j - \mathbf{s}_j \odot \mathbf{e}_{th})^{|\cdot|} \preceq \mathbf{w}_j \quad (1)$$

Where $\mathbf{e}_{xy} := (\mathbf{e}_x \parallel \mathbf{e}_y) \in \mathbb{R}^{2d}$ with \parallel representing concatenation and $e_x, e_y \in \mathbf{E}$. Furthermore, the inequality uses the following operators: the element-wise less or equal operator \preceq , the element-wise absolute value $\mathbf{x}^{|\cdot|}$ of a vector \mathbf{x} , and the element-wise (i.e., Hadamard) product \odot .

Scoring. SpeedE further enhances Min_SpeedE by adding the following two carefully designed scalar parameters to each relation embedding: (1) the inside distance slope $s_j^i \in [0, 1]$ and (2) the outside distance slope s_j^o with $s_j^i \leq s_j^o$. Let $m_j^i := 2s_j^i w + 1$, $m_j^o := 2s_j^o w + 1$, and

$k_j := m_j^o(m_j^o - 1)/2 - (m_j^i - 1)/(2m_j^i)$, then SpeedE defines the following distance function:

$$D(h, r_j, t) = \begin{cases} \tau_{\mathbf{r}_j(\mathbf{h}, \mathbf{t})} \oslash m_j^i, & \text{if } \tau_{\mathbf{r}_j(\mathbf{h}, \mathbf{t})} \leq w \\ \tau_{\mathbf{r}_j(\mathbf{h}, \mathbf{t})} \odot m_j^o - k_j, & \text{otherwise} \end{cases} \quad (2)$$

The distance function is separated into two piece-wise linear functions: (1) the inside distance $D_i(h, r_j, t) = \tau_{\mathbf{r}_j(\mathbf{h}, \mathbf{t})} \oslash m_j^i$ for triples that are captured to be true (i.e., $\tau_{\mathbf{r}_j(\mathbf{h}, \mathbf{t})} \leq w$) and (2) the outside distance $D_o(h, r_j, t) = \tau_{\mathbf{r}_j(\mathbf{h}, \mathbf{t})} \odot m_j^o - k_j$ for triples that are captured to be false (i.e., $\tau_{\mathbf{r}_j(\mathbf{h}, \mathbf{t})} > w$). Based on this function, SpeedE defines the score as $s(h, r_j, t) = -\|D(h, r_j, t)\|_2$. The intuition of s_j^i and s_j^o is that they control the slopes of the respective linear inside and outside distance functions. However, without any constraints on s_j^i and s_j^o , SpeedE would lose ExpressivE’s intuitive geometric interpretation [6] as s_j^i and s_j^o could be chosen in such a way that distances of embeddings within the hyper-parallelogram are larger than those outside. By constraining these parameters to $s_j^i \in [0, 1]$ and $s_j^i \leq s_j^o$, we preserve lower distances within hyper-parallelograms than outside and, thereby, the intuitive geometric interpretation of our embeddings.

4. Theoretical & Empirical Results

A gKGE’s inference capability is analyzed by studying which logical rules it captures. The set of core logical rules, commonly studied in the gKGE literature [7, 16, 6], consists of (1) symmetry $r_1(X, Y) \Rightarrow r_1(Y, X)$, (2) anti-symmetry $r_1(X, Y) \wedge r_1(Y, X) \Rightarrow \perp$, (3) inversion $r_1(X, Y) \Leftrightarrow r_2(Y, X)$, (4) composition $r_1(X, Y) \wedge r_2(Y, Z) \Rightarrow r_3(X, Z)$, (5) hierarchy $r_1(X, Y) \Rightarrow r_2(X, Y)$, (6) intersection $r_1(X, Y) \wedge r_2(X, Y) \Rightarrow r_3(X, Y)$, and (7) mutual exclusion $r_1(X, Y) \wedge r_2(X, Y) \Rightarrow \perp$. Surprisingly, we find in Theorem 4.1 that SpeedE still captures all core logical rules (see [1], Appendix H).

Theorem 4.1. *SpeedE captures the set of core logical rules.*

Inference Time. The most costly operations during inference are operations on vectors. Thus, we can estimate ExpressivE’s and SpeedE’s inference time by counting the number of vector operations necessary for computing a triple’s score: By reducing the width vector to a scalar, many operations reduce from a vector to a scalar operation. In particular, ExpressivE needs 15, whereas SpeedE needs solely 8 vector operations to compute a triple’s score. In [1], we empirically measure the inference time of SpeedE, ExpressivE, RotH, and AttH under the same parameter configurations on each benchmark, finding that SpeedE halves ExpressivE’s inference time as expected and even solely requires about a sixth of RotH’s and AttH’s inference time.

KGC Results. Following [11], we evaluate each gKGE’s performance under low dimensionalities with $d = 32$. Table 1 reports their MRR and H@1 scores (for the complete results, see [1]). It reveals that on YAGO3-10 – the largest benchmark – SpeedE outperforms any SotA gKGE by a relative difference of 7% on H@1, providing strong evidence for SpeedE’s scalability to large KGs. Furthermore, it shows that our enhanced SpeedE model is competitive with SotA gKGEs on FB15k-237 and even outperforms any competing gKGE on WN18RR by a large margin.

Convergence Time & Model Size. To quantify the convergence time, we measure for each gKGE the time to reach a validation MRR score of 0.490, i.e., approximately 1% less than the worst reported MRR score of Table 2. As outlined in Table 2, SpeedE converges already after 6min.

Table 1Low-dimensional ($d = 32$) KGC results of SotA gKGEs on WN18RR, FB15k-237, and YAGO3-10.

Model		WN18RR		FB15k-237		YAGO3-10	
		MRR	H@1	MRR	H@1	MRR	H@1
Euclidean	SpeedE	.493	.446	.320	.227	.413	.332
	Min_SpeedE	.485	.442	.319	.226	.410	.328
	ExpressivE	.485	.442	.298	.208	.333	.257
	TuckER [10]	.428	.401	.306	.223	-	-
Non-Euclid.	RefH [11]	.447	.408	.312	.224	.381	.302
	RotH [11]	.472	.428	.314	.223	.393	.307
	AttH [11]	.466	.419	.324	.236	.397	.310
	ConE [13]	.471	.436	-	-	-	-

Table 2

Dimensionality, MRR, convergence time, and number of parameters of SotA gKGE’s on WN18RR.

Model	Dim.	MRR	Conv. Time	#Parameters
SpeedE	50	.500	6min	2M
ExpressivE	200	.500	31min	8M
ConE	500	.496	1.5h	20M
RotH	500	.496	2h	21M

Thus, while keeping strong KGC performance on WN18RR, SpeedE speeds up ExpressivE’s convergence time by a factor of 5, ConE’s by 15, and RotH’s by 20. Furthermore, the table shows that SpeedE ($d = 50$) needs solely a quarter of ExpressivE’s ($d = 200$) and a tenth of ConE’s and RotH’s ($d = 500$) parameters to achieve a similar or slightly better KGC performance.

5. Conclusion

In this work, we introduce SpeedE, a lightweight gKGE that (1) captures the set of core logical rules, (2) is competitive with SotA gKGEs, even significantly outperforming them on YAGO3-10 and WN18RR, and (3) dramatically increases the efficiency of current gKGEs, needing solely a fifth of the training time and a fourth of the number of parameters of the SotA ExpressivE model on WN18RR to reach the same KGC performance. To facilitate the reproducibility of our results and the use of our model, we provide SpeedE’s code in a public GitHub repository¹.

Acknowledgements

Financial support for this research has been provided by the Vienna Science and Technology Fund (WWTF) under grants [10.47379/VRG18013, 10.47379/NXT22018, 10.47379/ICT2201], as well as the Christian Doppler Research Association (CDG) JRC LIVE.

¹<https://github.com/AleksVap/SpeedE>

References

- [1] A. Pavlović, E. Sallinger, SpeedE: Euclidean geometric knowledge graph embedding strikes back, in: K. Duh, H. Gomez, S. Bethard (Eds.), Findings of the Association for Computational Linguistics: NAACL 2024, Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 69–92. URL: <https://aclanthology.org/2024.findings-naacl.6>.
- [2] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, *IEEE Trans. Knowl. Data Eng.* 29 (2017) 2724–2743. URL: <https://doi.org/10.1109/TKDE.2017.2754499>. doi:10.1109/TKDE.2017.2754499.
- [3] S. Broscheit, K. Gashteovski, Y. Wang, R. Gemulla, Can we predict new facts with open knowledge graph embeddings? A benchmark for open link prediction, in: D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020, pp. 2296–2308. URL: <https://doi.org/10.18653/v1/2020.acl-main.209>. doi:10.18653/v1/2020.acl-main.209.
- [4] A. Pavlović, E. Sallinger, Building bridges: Knowledge graph embeddings respecting logical rules (short paper), in: B. Kimelfeld, M. V. Martinez, R. Angles (Eds.), Proceedings of the 15th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW 2023), Santiago de Chile, Chile, May 22-26, 2023, volume 3409 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: <https://ceur-ws.org/Vol-3409/paper9.pdf>.
- [5] A. Pavlović, E. Sallinger, Expressive and geometrically interpretable knowledge graph embedding (extended abstract), in: The First Austrian Symposium on AI, Robotics, and Vision (AIROV24), 2024. URL: <https://semantic-systems.org/sites/KG-NeSy/papers/P60.pdf>.
- [6] A. Pavlović, E. Sallinger, ExpressivE: A spatio-functional embedding for knowledge graph completion, in: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, 2023. URL: https://openreview.net/pdf?id=xkev3_np08z.
- [7] 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, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019. URL: <https://openreview.net/forum?id=HkgEQnRqYQ>.
- [8] 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: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 2731–2741. URL: <https://proceedings.neurips.cc/paper/2019/hash/d961e9f236177d65d21100592edb0769-Abstract.html>.
- [9] Z. Cao, Q. Xu, Z. Yang, X. Cao, Q. Huang, Dual quaternion knowledge graph embeddings, *Proceedings of the AAAI Conference on Artificial Intelligence* 35 (2021) 6894–6902. URL: <https://doi.org/10.1609/aaai.v35i8.16850>. doi:10.1609/aaai.v35i8.16850.
- [10] K. Wang, Y. Liu, D. Lin, M. Sheng, Hyperbolic geometry is not necessary: Lightweight Euclidean-based models for low-dimensional knowledge graph embeddings, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computa-

- tional Linguistics, Punta Cana, Dominican Republic, 2021, pp. 464–474. URL: <https://doi.org/10.18653/v1/2021.findings-emnlp.42>. doi:10.18653/v1/2021.findings-emnlp.42.
- [11] I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi, C. Ré, Low-dimensional hyperbolic knowledge graph embeddings, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 6901–6914. URL: <https://doi.org/10.18653/v1/2020.acl-main.617>. doi:10.18653/v1/2020.acl-main.617.
- [12] I. Balazevic, C. Allen, T. M. Hospedales, Multi-relational poincaré 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: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 4465–4475. URL: <https://proceedings.neurips.cc/paper/2019/hash/f8b932c70d0b2e6bf071729a4fa68dfc-Abstract.html>.
- [13] Y. Bai, Z. Ying, H. Ren, J. Leskovec, Modeling heterogeneous hierarchies with relation-specific hyperbolic cones, in: M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, 2021, pp. 12316–12327. URL: <https://proceedings.neurips.cc/paper/2021/hash/662a2e96162905620397b19c9d249781-Abstract.html>.
- [14] 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: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 2018, pp. 4289–4300. URL: <https://proceedings.neurips.cc/paper/2018/hash/b2ab001909a8a6f04b51920306046ce5-Abstract.html>.
- [15] Z. Zhang, J. Cai, Y. Zhang, J. Wang, Learning hierarchy-aware knowledge graph embeddings for link prediction, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, AAAI Press, 2020, pp. 3065–3072. URL: <https://doi.org/10.1609/aaai.v34i03.5701>. doi:10.1609/aaai.v34i03.5701.
- [16] R. Abboud, Í. Í. 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: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL: <https://proceedings.neurips.cc/paper/2020/hash/6dbbe6abe5f14af882ff977fc3f35501-Abstract.html>.