# Towards General Geometries for Embedding Knowledge Graphs

**Samuel G. Fadel** [1] **Tino Paulsen** [2] **Sebastian Mair** [3]

## Abstract

When embedding knowledge graphs, choosing the right geometry can heavily impact the expressivity of the embedding model to accurately predict the relations of the knowledge graph. Importantly, the structure of the chosen space should ideally accommodate the structure in the graph. Existing approaches describe embeddings on non-Euclidean geometries by closed-form analytical equations describing movements on them, limiting the geometry of choice to canonical cases; others forego this by reformulating the problem in a way that removes an interpretation grounded on geometry. In this paper, we generalise the conceptualisation of learning embeddings on manifolds by allowing any choice of metric to be used, whether for which known closed-form equations for movement are known or not. Experimentally, we show that this not only recovers existing approaches, but highlights the practicality of learning embeddings on general geometries.

## 1. Introduction

Knowledge graph (KG) embedding is the problem of learning representations for nodes (or entities) and relations between them, aiming to enable automated reasoning about facts collected about a specific subject. By casting this problem as a link prediction task, existing methods walk fine lines between expressivity, interpretability, and overfitting. Often, it is trivial for models to fit the training data, but generalisation is challenging due to differentiating unobserved links from non-existent links. Thus, many successful approaches aim for modestly-sized models built on sophisticated mathematical frameworks. Examples include exploiting properties of the complex field or leveraging geometric properties under metrics different from Euclidean. Despite subsuming a lot of previous methods, recent approaches such as FieldE (Nayyeri et al., 2021) are built for using specific manifolds, but require explicit closed-form equations for moving points on that manifold. Ideally, using arbitrary manifolds without having the need of knowing such closed-form equations is more convenient and flexible.

In this paper, we propose Manifold Embeddings (ManE), which uses arbitrary manifolds for KG embedding while not requiring explicit closed-form equations for moving points on that manifold. Instead, only the desired manifold metric tensor has to be defined. This is critical as it also allows for mixing metrics and thus, mixing manifolds and their geometric properties. We introduce the formulation without explicit closed-form equations and evaluate our proposed approach on Poincaré disks, spheres, and mixtures thereof.

**Related work.** Earlier work on embedding KGs introduced modern conceptualisations that leverage positioning and movement in space as a way to convey the underlying connectivity given by the graph. TransE (Bordes et al., 2013) models relations as learnable vectors that are added to vector representations of entities, where the idea is that relations place connected nodes close to each other. DistMult (Yang et al., 2015) attempts a similar setting, but instead models the relation as a bilinear transformation. While simple, both approaches suffer from a lack of capacity in representing all possible patterns of relations. Aiming to increase expressivity while maintaining the intituition of previous approaches, RotatE (Sun et al., 2019) and QuatE (Zhang et al., 2019) extend embeddings to complex (and quaternion) spaces, where the use of rotations represented as products therein increase the expressivity of the embedding and link prediction model, at the cost of usability of learned embeddings in downstream tasks.

Different geometries should be preferred for different structures of datasets, like hyperbolic geometry showing capabilities to embed hierarchical structures (Mathieu et al., 2019), while spherical geometry serves data with more cyclical nature well (Davidson et al., 2018). Skopek et al. (2020) formulated a mixed-curvature VAE which is achieved through a combination of manifolds as components of constant curvature. The representations are decomposed in parts, processed with operators and concatenated again. In contrast, we directly calculate on the manifold.

[1] Linköping University, Sweden [2] Leuphana University of Lüneburg, Germany [3] Uppsala University, Sweden . Correspondence to: Samuel G. Fadel <samuel@nihil.ws>.

*Proceedings of the Geometry-grounded Representation Learning and Generative Modeling at $41^{st}$ International Conference on Machine Learning*, Vienna, Austria. PMLR Vol Number, 2024.

## 2. Preliminaries

**Knowledge graphs.** A knowledge graph (KG) is a set of tuples $(u, r, w)$, such that $r \in \mathcal{R}$ is a relation type and $u, w \in \mathcal{E}$ are entities. The set $\mathcal{R}$ is comprised of symbols such as `parent-of` and `sibling-of`, while $\mathcal{E}$ is comprised of symbols such as `PersonA` and `PersonB`. Our goal is to learn a model/representation that allows us to query the knowledge represented in the graph. For instance, given $(\texttt{PersonC}, \texttt{parent-of}, \texttt{PersonA})$ and $(\texttt{PersonC}, \texttt{parent-of}, \texttt{PersonB})$ are in the knowledge graph, is $(\texttt{PersonA}, \texttt{sibling-of}, \texttt{PersonB})$ true? A similar task is to find the most likely candidates for completing the tuple $(\texttt{PersonA}, \texttt{sibling-of}, \cdot)$.

Organising facts this way provides an opportunity for solving learning problems that require reasoning from facts about a certain domain. A representation of this data which is useful for machines to learn from is thus a requirement to achieve this goal. Previous research (Mathieu et al., 2019; Nagano et al., 2019; Chami et al., 2020; Nayyeri et al., 2021) has shown that graphs have different (sub-)structures which originate from the way its nodes are connected, making the notion of closeness in the graph and the embeddings of certain geometries a better match for each other. For instance, tree-structures are believed to be more appropriately represented in hyperbolic spaces due to the exponential growth in the number of nodes away from the root node, just as the volume in hyperbolic spaces grows exponentially as points are further away from the origin.

Solving this problem thus requires a formulation with the necessary flexibility to encompass any desired geometry in the embedding process, while still enabling intervention by choosing specific manifolds using domain knowledge as deemed necessary.

**Riemannian geometry.** We introduce selected concepts from Riemannian geometry which will be needed, note that this is not exhaustive. For a more complete introduction we recommend Lee (2018). Given a smooth manifold $\mathcal{M}$ and a point $\boldsymbol{z} \in \mathcal{M}$, $\mathcal{T}_{\boldsymbol{z}}\mathcal{M}$ is the associated tangent space of point $\boldsymbol{z}$, which is a vector space containing all tangent vectors. The disjoint union of all tangent spaces $\mathcal{T}_{\boldsymbol{z}}\mathcal{M}$ over all points of the manifold is considered the tangent bundle $\mathcal{T}\mathcal{M}$. A Riemannian manifold $(\mathcal{M}, \mathfrak{g}_{\boldsymbol{z}})$ is a smooth manifold equipped with a Riemannian metric $\mathfrak{g}_{\boldsymbol{z}}$, that is a inner product changing smoothly with point $\boldsymbol{z} \in \mathcal{M}$ :

$$\mathfrak{g}_{\boldsymbol{z}} = \langle \cdot, \cdot \rangle_{\boldsymbol{z}} : \mathcal{T}_{\boldsymbol{z}}\mathcal{M} \times \mathcal{T}_{\boldsymbol{z}}\mathcal{M} \to \mathbb{R}.$$

The Riemannian metric is sufficient to describe the local geometry, from which ideas from global geometry can be constructed. From the inner product a norm on the tangent space $\mathcal{T}_{\boldsymbol{z}}\mathcal{M}$ follows, given by $\|\cdot\|_{\boldsymbol{z}} = \sqrt{\langle \cdot, \cdot \rangle_{\boldsymbol{z}}}$. Often the Riemannian metric $\mathfrak{g}_{\boldsymbol{z}}$ is rewritten into a matrix representation $G(\boldsymbol{z})$, called the metric tensor:

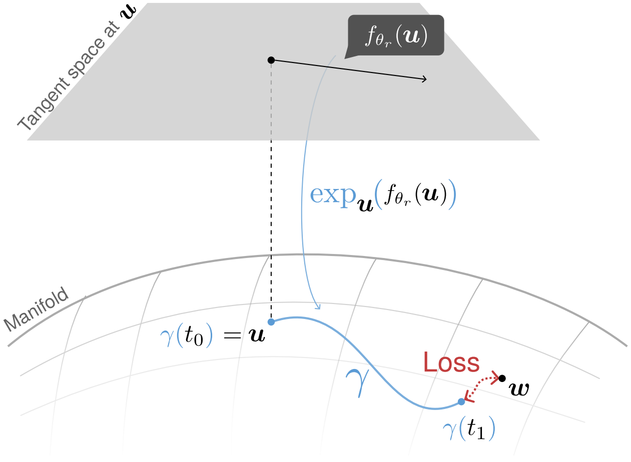
Tangent space at $\boldsymbol{u}$
$f_{\theta_r}(\boldsymbol{u})$
$\exp_{\boldsymbol{u}}(f_{\theta_r}(\boldsymbol{u}))$
Manifold
$\gamma(t_0) = \boldsymbol{u}$
$\gamma$
Loss
$\boldsymbol{w}$
$\gamma(t_1)$

*Figure 1.* In algorithmic order: At an embedded point $\boldsymbol{u}$ in the manifold the function $f_{\theta_r}(\boldsymbol{u})$ produces a velocity which serves as input for the exponential map. The execution of the exponential map follows a geodesic $\gamma$, whose endpoint $\gamma(t_1)$ is then compared to point $\boldsymbol{w}$ using a loss.

$$\forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{T}_{\boldsymbol{z}}\mathcal{M},\ \langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\boldsymbol{z}} = \mathfrak{g}(\boldsymbol{z})(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^{\intercal} G(\boldsymbol{z}) \boldsymbol{v}.$$

Using this metric tensor, a measure is typically given by $\mathrm{d}\mathcal{M}(\boldsymbol{z}) = \sqrt{|G(\boldsymbol{z})|}\mathrm{d}\boldsymbol{z}$ with $\mathrm{d}\boldsymbol{z}$ being the Lebesgue measure. With all this machinery, one can start the search for the shortest-length path between two points $\boldsymbol{z}, \boldsymbol{y} \in \mathcal{M}$. Given a curve $\gamma : t \mapsto \gamma(t) \in \mathcal{M}$, its length is given by

$$L(\gamma) = \int_0^1 \|\gamma'(t)\|^{1/2}_{\gamma(t)}.$$

The shortest path is then pragmatically taken as $\gamma^* = \arg\min_{\gamma} L(\gamma)$ with $\gamma(0) = \boldsymbol{z}$ and $\gamma(1) = \boldsymbol{y}$, resulting in the geodesic from $\boldsymbol{z}$ to $\boldsymbol{y}$. Measuring the length of this shortest path can be used to define a global distance $d_{\mathcal{M}}(\boldsymbol{z}, \boldsymbol{y}) = \inf L(\gamma)$. Given a point $\boldsymbol{z} \in \mathcal{M}$ and a velocity $\boldsymbol{v} \in \mathcal{T}_{\boldsymbol{z}}\mathcal{M}$, the operator for executing the movement on the manifold indicated by that velocity is called the exponential map $\exp_{\boldsymbol{z}}(\boldsymbol{v})$. The exponential map results in following the geodesic according to the velocity and returns the endpoint of the geodesic. If the exponential map is defined for every point $\boldsymbol{z} \in \mathcal{M}$, then $\mathcal{M}$ is geodesically complete.

Finding geodesics is analytically solved for some manifolds, for exploring new manifolds the analytical method is often not feasible. We resort to numerically approximating the exponential map with a differential equation (do Carmo & Flaherty Francis, 1992).

## 3. Method

We propose an approach to the problem which allows models to use any geodesically complete Riemannian manifold that can be practically described by their metric tensor.

We aim to minimise the per-point loss given by

$$-\Big[\log\sigma(\xi - \operatorname{diss}(f_{\theta_r}(h), t)) + \sum_{j\in\mathcal{N}} \log\sigma(\operatorname{diss}(f_{\theta_{r'_j}}(h'_j), t'_j) - \xi)\Big],$$

where $\xi \geq 0$ is a margin, $\mathcal{N}$ the set of negative examples derived from the tuple $(h, r, t)$, $\sigma$ is the sigmoid function, and $\operatorname{diss}(\cdot,\cdot)$ is a dissimilarity measure one can utilise any appropriate choice of distance measure. This could change depending on the manifold or be approximated by, e.g., the Euclidean distance, as long as the distance only rescales and does not change the optimum. In that case one has to expect a slowing effect on training at least, as the gradients will not match the proper size.

This conceptualisation can be understood as a refinement of distance-based embeddings. There, embeddings are placed closer or further away according to some criteria, with the goal of making similar objects close, while dissimilar ones are further away. Here, the information provided by relations between entities allows us to instead think of it in terms of *reachability under a certain relation*. Thus, for each relation type $r \in \mathcal{R}$, we learn a function $f_{\theta_r}$ that takes an entity embedding and moves it close to the embeddings of entities it should be connected to.

Parameter sharing happens per relation, where each $\theta_r$ parameterise the velocities at each point in time according to the appropriate relation type $r$. This means that, for instance, when node $u$ appears in other relations, one can end up in different end points by following different velocity functions (different relation types $r$).

We model a relation by parameterising the velocity of a trajectory $\gamma$ going from a starting point $\gamma(t_0) = \boldsymbol{u}$ (embedding of $u$) to end up at the proper end point $\gamma(t_1) = \boldsymbol{w}$ (embedding of $w$). Such a trajectory represents a notion of "connectivity", whereby entities should be reachable by each other if a relation exists between them.

We define those paths indirectly, instead learning functions that describe the velocity at each point in the trajectory. Mathematically, we model this as the following initial value problem (IVP)

$$f_{\theta_r}(\gamma(t)) = \frac{\partial\gamma(t)}{\partial t}, \quad \text{with } \gamma(t_0) = \boldsymbol{u},$$

where we attempt to ensure that the solution is as close to $\boldsymbol{w}$ as possible (with canonical choices $t_0 = 0$ and $t_1 = 1$). That is, by following the path $\gamma$ starting at $\boldsymbol{u}$, we parametrise and learn some $f_{\theta_r}$ that ends up as close to $\boldsymbol{w}$ as possible, if $(u, r, w)$ is an observed tuple.

We interpret the solution to the IVP as the exponential map $\exp_{\boldsymbol{u}}(\boldsymbol{v}) = \gamma(t_1)$, while $f_{\theta_r}(\gamma(t))$ represents a point in the tangent space of any point in the trajectory $\gamma$. As an important consequence, this means that we model how we navigate the space where the nodes are embedded by changing the solutions to the IVP. In other words, if we describe the solution to the IVP in terms of the metric tensor of a Riemannian manifold, by changing the metric tensor, we change the solutions and therefore implicitly learn a different geometry. In contrast to previous work (Nayyeri et al., 2021), our approach does not explicitly evoke or impose a choice of geometry only for cases where analytical solutions are known. Note that without this limitation, one is free to also parameterise the metric tensor itself, as we show in an example. The method is summarised in Figure 1.

Even though the choice of manifold is free, we have to choose a quite finite number for experiments. For comparability, we choose the Euclidean and hyperbolic geometries. Spherical geometry was indicated by Nayyeri et al. (2021), but not shown, Euclidean and hyperbolic geometry can be compared. Hyperbolic geometry can be constructed by different models, we choose the Poincaré ball model and denote it by $\mathbb{B}_c = (\mathcal{B}_c, \mathfrak{g}_b^c)$ with $\mathcal{B}_c$ being the open ball of radius $1/\sqrt{c}$ and $\mathfrak{g}_b^c$, the Poincaré ball metric tensor, given by

$$\mathfrak{g}_b^c(\boldsymbol{z}) = (\lambda_{\boldsymbol{z}}^c)^2\, \mathfrak{g}_e(\boldsymbol{z}), \quad \lambda_{\boldsymbol{z}}^c = \frac{2}{1 - c\|\boldsymbol{z}\|^2},$$

where $\lambda_{\boldsymbol{z}}^c$ is the conformal factor and $\mathfrak{g}_e$ is the Euclidean metric tensor, that is the usual dot product. $c$ is the curvature, which for the Poincaré ball is a constant negative number.

Spherical geometry is a constantly positive curved space which is often defined in terms of a radius, as this geometry can be comfortably embedded into a Euclidean space of dimension $n + 1$. We follow the Riemannian metric given by Lee (2018) with a radius $r$:

$$r^2\mathfrak{g}_s(x/r) = \frac{4r^4(\mathrm{d}x_1^2 + \ldots + \mathrm{d}x_n^2)}{(||x||^2 + r^2)^2}.$$

Note that this is a stereographic projection, as well as the Poincaré ball.

We utilise that fact in combining both metrics using a convex combination on the metrics we denote by $\mathfrak{g}_\varphi$:

$$\mathfrak{g}_\varphi = a \cdot \mathfrak{g}_b^c + (1 - a)\mathfrak{g}_s^c$$

with $0 \leq a \leq 1$ as a learnable parameter. A consequence of $a$ being learnable is that the model can decide which kind of geometry it prefers for the problem at hand. The resulting exponential map is, to the best of our knowledge, not analytically known and only feasible to solve for numerically.

*Table 1.* Performances of several models on two real-world data sets. Note that FieldP and ManE-$\mathbb{B}$ both use a Poincaré ball as a manifold. The FieldS row includes no numbers as it was not demonstrated by the original authors.

| | WN18RR | | | | FB15k-237 | | | |
|---|---|---|---|---|---|---|---|---|
| | MRR | Hit@1 | Hit@3 | Hit@10 | MRR | Hit@1 | Hit@3 | Hit@10 |
| FieldE | 0.3455 | 0.2892 | 0.3789 | 0.4403 | 0.1711 | 0.1071 | 0.1907 | 0.3014 |
| ManE | 0.3448 | 0.2862 | 0.3816 | 0.4432 | 0.1736 | 0.1120 | 0.1934 | 0.2957 |
| FieldP | 0.0700 | 0.0284 | 0.0702 | 0.1619 | 0.1255 | 0.0891 | 0.1327 | 0.1951 |
| ManE-$\mathbb{B}$ | 0.0247 | 0.0148 | 0.0272 | 0.0415 | 0.1196 | 0.0870 | 0.1234 | 0.1834 |
| FieldS | - | - | - | - | - | - | - | - |
| ManE-$\mathbb{S}$ | 0.1375 | 0.0722 | 0.1564 | 0.2815 | 0.1679 | 0.1058 | 0.1798 | 0.2927 |
| ManE-MM | 0.0191 | 0.0005 | 0.0211 | 0.0511 | 0.0697 | 0.0409 | 0.0710 | 0.1274 |

## 4. Experiments

We experimentally evaluate our proposed ManE model using the standard knowledge graph link prediction task.

**Baselines.** The most relevant baseline to compare against is FieldE (Nayyeri et al., 2021) which supports Euclidean geometry (FieldE) and hyperbolic geometry on the Poincaré ball (FieldP). Note that a version using spherical geometry (FieldS) was not demonstrated by the authors. The baselines are reimplemented using the authors' code as a reference.

**Data.** To evaluate ManE, we use two real-world data sets: FB15k-237 (Toutanova & Chen, 2015) and WN18RR (Dettmers et al., 2018).

**Metrics.** All models get evaluated using standard metrics for knowledge graph completion, namely Mean Reciprocal Rank (MRR) and Hits@$k$, with $k \in \{1, 3, 10\}$. Both metrics are in their filtered variants.

**Setup.** We train each model for 1000 epochs, stop early if the validation MRR does not improve for 50 epochs and pick the model with the best validation MRR among all epochs. Furthermore, we use 50 negative samples using the same sampling as Wang et al. (2014), use $d = 32$ as an embedding size, and perform a grid search over margins $\xi \in \{1.0, 5.0\}$ and learning rates $\{1\mathrm{e}{-2}, 1\mathrm{e}{-3}\}$. Also, we use Xavier to initialise the embeddings. Following Nayyeri et al. (2021), we model the inverse relations as a second set of relations.

The function $f_{\theta_r} : \mathbb{R}^d \to \mathbb{R}^d$ is parameterised per relation $r$ using a two-layer neural network that has in its hidden layer a dimensionality of $d_{\text{hidden}} = 32$ and $\tanh$ as an activation function. The output layer has no activation function.

As an optimiser, we use ADAM with weight decay (Loshchilov & Hutter, 2019), since our parameters are all defined on the tangent bundle. Following the same logic, our choice of $\mathrm{diss}(\cdot)$ is the standard Euclidean distance. Per model, no suffix denotes the Euclidean space, -$\mathbb{B}$ the Poincaré ball, and -$\mathbb{S}$ a hypersphere. For ManE, we also evaluate a mixed metric approach suffixed with -MM. Utilising numerical optimisation of the exponential map allows for an arbitrary choice of manifold, which we evaluate here as a learned convex combination of the Poincaré ball and spherical metrics.

**Results.** The results are presented in Table 1. In the Euclidean setting, our model achieves comparable performance as FieldE on both datasets. Using the Poincaré ball, both the baseline and our model demonstrate similar behaviour on FB15k-237, and both underperform on WN18RR. The spherical geometry shows promising performance, in contrast the mixed metric model still needs improvement.

## 5. Discussion and conclusion

We generalised FieldE with regard to the choice of manifold, allowing arbitrary metrics to be used by choosing to approximate the exponential map numerically. The velocity generator $f_{\theta_r}$ learns a common handling per relation $r \in \mathcal{R}$, which allows for flexible modeling of each relation. Exploring new manifolds allows the model to fit the structure of the data, potentially increasing expressivity of the models. Empirically, we show our formulation does not fall short on performance, while still allowing the aforementioned flexibility. As an attempt at showcasing this, we experimented with a model using a mixed metric, using both hyperbolic and spherical metrics.

However, the gained flexibility comes at a cost. For example, on WN18RR FieldE took 0.176 minutes per epoch (min/ep) while ManE took 0.180 min/ep. On the same data, FieldP took 0.32 min/ep and ManE-$\mathbb{B}$ took 10.9 min/ep. As we can see, optimisation is an open problem. Using an Euclidean distance in the loss does not approximate well the distance on the manifold in all cases, potentially introducing numerical instability, which seems to be more prevalent with the Poincaré ball metric used. Optimising on general manifolds is desirable, but the price to pay during training is still high.

## Acknowledgements

SF is financially supported by the Excellence Center at Linköping–Lund in Information Technology (ELLIIT). TP is supported by the German Federal Ministry of Education and Research (BMBF) under the project 16KIS1205. SM is partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation; as well as Sweden's Innovation Agency (Vinnova) project 2022-03023. The computations were enabled by the Berzelius resource provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre.

## References

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Chami, I., Wolf, A., Juan, D.-C., Sala, F., Ravi, S., and Ré, C. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6901–6914, 2020.

Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. Hyperspherical variational auto-encoders. In *Conference on Uncertainty in Artificial Intelligence*, 2018.

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

do Carmo, M. P. and Flaherty Francis, J. *Riemannian geometry*, volume 2. Springer, 1992.

Lee, J. M. *Introduction to Riemannian Manifolds*. Springer, 2 edition, 2018.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Mathieu, E., Le Lan, C., Maddison, C. J., Tomioka, R., and Teh, Y. W. Continuous hierarchical representations with Poincaré variational auto-encoders. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Nagano, Y., Yamaguchi, S., Fujita, Y., and Koyama, M. A wrapped normal distribution on hyperbolic space for gradient-based learning. In *International Conference on Machine Learning*, pp. 4693–4702. PMLR, 2019.

Nayyeri, M., Xu, C., Hoffmann, F., Alam, M. M., Lehmann, J., and Vahdati, S. Knowledge graph representation learning using ordinary differential equations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 9529–9548, 2021.

Skopek, O., Ganea, O.-E., and Bécigneul, G. Mixed-curvature variational autoencoders. In *International Conference on Learning Representations*, 2020.

Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019.

Toutanova, K. and Chen, D. 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.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

Yang, B., Yih, S. W.-t., He, X., Gao, J., and Deng, L. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations*, 2015.

Zhang, S., Tay, Y., Yao, L., and Liu, Q. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems*, volume 32, 2019.