Less Is MuRE: Revisiting Shallow Knowledge Graph Embeddings

Victor Charpenay
Mines Saint-Etienne, UMR 6158 LIMOS
Saint-Étienne, France
victor.charpenay@emse.fr

Steven Schockaert
Cardiff University
Cardiff, UK
schockaerts1@cardiff.ac.uk

Abstract

In recent years, the field of knowledge graph completion has focused on increasingly sophisticated models, which perform well on link prediction tasks, but are less scalable than earlier methods and are not suitable for learning entity embeddings. As a result, shallow models such as TransE and ComplEx remain the most popular choice in many settings. However, the strengths and limitations of such models remain poorly understood. In this paper, we present a unifying framework and systematically analyze a number of variants and extensions of existing shallow models, empirically showing that MuRE and its extension, ExpressivE, are highly competitive. Motivated by the strong empirical results of MuRE, we also theoretically analyze the expressivity of its associated scoring function, surprisingly finding that it can capture the same class of rule bases as state-of-the-art region-based embedding models.

1 Introduction

Knowledge graphs (KGs) encode knowledge in the form of subject-predicate-object triples, such as $(\textit{Paris}, \textit{capital-of}, \textit{France})$. Within NLP, such resources can help to address some of the limitations of Large Language Models (LLMs), for instance by providing knowledge about rare entities, which LLMs are known to struggle with (Mallen et al., 2023; Kandpal et al., 2023; Huang et al., 2024), and by providing domain-specific and up-to-date knowledge more generally. However, KGs are notoriously incomplete, which has prompted a plethora of methods for predicting plausible missing triples. Early work on this topic was dominated by *shallow* KG embedding models, where plausibility scores are simple and efficient to compute, typically having a linear or bilinear form (Bordes et al., 2013; Yang et al., 2015; Trouillon et al., 2017; Balazevic et al., 2019a). In recent years, alternative approaches have emerged, based on Graph Neural Networks (Zhu et al., 2021), pre-trained language models (Yao et al., 2019; Lv et al., 2022) and LLMs (Zhang et al., 2024; Wei et al., 2023). These methods generally lead to more accurate link prediction results, but at the cost of reduced scalability. Yet, in downstream tasks, including alignment with language models (Zhang et al., 2019b; Wang et al., 2021), entity retrieval for retrieval-augmented generation (Matsumoto et al., 2024; Doan et al., 2024) and recommendations (Wang et al., 2019), scalability is often more crucial than accuracy.

Shallow KG embedding models are efficient, easy to implement, and often perform sufficiently well, but their relative strengths and weaknesses remain poorly understood. Existing models tend to differ in more than one aspect, making it unclear how different design choices affect their performance. In this paper, we aim to further our understanding of shallow KG embedding models by making two contributions. First, we theoretically analyze a variant of MuRE (Balazevic et al., 2019a). MuRE can essentially be seen as a natural generalization of two of the most popular models, namely TransE (Bordes et al., 2013) and DistMult (Yang et al., 2015), and thus serves as a natural starting point for studying shallow embeddings. We show that MuRE overcomes the theoretical limitations of TransE and DistMult. In fact, we find that its theoretical expressivity, surprisingly, matches what is known for more complex KG embedding models, which have been specifically designed with expressivity in mind (Abboud et al., 2020; Pavlovic and Sallinger, 2023; Charpenay and Schockaert, 2024).

Second, we carry out a systematic empirical analysis, to better understand how different design choices affect the performance of shallow embedding models. We find that MuRE is highly competitive among existing shallow embedding models. However, for some benchmarks and some evaluation metrics, its performance can be improved by incorporating design choices from other models,

such as the addition of cross-coordinate comparisons, which we borrow from ComplEx (Trouillon et al., 2017) and BoxE (Abboud et al., 2020).

2 Related Work

In this paper, we establish connections between different KG embedding models, a task which has already received considerable attention. For example, RESCAL (Nickel et al., 2011a) was proven to be a special case of the concurrent Neural Tensor Network model (Nickel et al., 2016a). HolE (Nickel et al., 2016b), a model with real-valued embeddings, was shown to be equivalent to ComplEx (Trouillon et al., 2017), a generalization of DistMult in complex space (Trouillon and Nickel, 2017; Liu et al., 2017). Interestingly, the two models nonetheless behave differently during training, favouring ComplEx in practice. TuckER (Balazevic et al., 2019b) has been designed as a generalization of DistMult and ComplEx. Conversely, SimplE (Kazemi and Poole, 2018) was designed as a simplification of ComplEx. Despite these convergence attempts, shallow KG embeddings models continue to be seen as belonging to two separate families: bilinear models—thought of as tensor factorization approaches—and linear models—also called distance-based models. In fact, MuRE bridges the gap between these two families, but recent surveys of KG embedding models (Hogan et al., 2021; Ali et al., 2022) still maintain this divide. In the next section, we will build on the insight that MuRE generalizes both linear and bilinear models to provide a unifying framework for studying shallow KG embedding models.

Several attempts at revisiting KG embedding baselines have already been made. Kadlec et al. (2017) showed that the performance of DistMult could be improved by using a more sophisticated loss function. Later, Ruffinelli et al. (2020) reevaluated five models in various settings, highlighting the strong performance of RESCAL. A similar effort with 21 models showed that MuRE, RotatE and, to a lesser extent, TransE were more robust than other models (Ali et al., 2022). The two latter studies are based on the idea that a single model can be trained under many different configurations, which often leads to improvements over the best known results for that model. In this paper, instead of exploring novel training configurations, we look at novel scoring functions, based on a core set of features that existing models have in common.

Our study unifies some of the best-known KG embedding models. However, we do not aim to capture all possible shallow models. For instance, in our main analysis, we exclude QuatE, which generalizes ComplEx with quaternion embeddings (Zhang et al., 2019a), RotatE (Sun et al., 2019), which models relations as rotations in Euler planes, and hyperbolic models such as MuRP (Balazevic et al., 2019a) and AttH (Chami et al., 2020). Because of the great diversity of shallow models found in the literature, other classifications have emerged (Cao et al., 2024), and there have been other attempts to unify classes of shallow models (Yang and Liu, 2021). Our proposal based on MuRE has been driven both by empirical experience and theoretical expressivity results.

3 Shallow Embedding Models

Preliminaries Let $\mathcal{E}$ be a set of entities and $\mathcal{R}$ a set of relations. A KG is a set of triples of the form $\mathcal{G} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. Intuitively, the triple (e, r, f) encodes the fact that entity e (head entity) is related to entity f (tail entity) through relation r. In shallow KG embeddings, both entities and relations are represented as vectors. A shallow embedding model is parameterized by an entity embedding, which maps each entity $e \in \mathcal{E}$ to a corresponding vector $\mathbf{e} \in \mathbb{R}^n$, and a relation embedding, which maps each relation $r \in \mathcal{R}$ to a corresponding vector $\mathbf{r} \in \mathbb{R}^m$. We refer to n as the dimensionality of the KG embedding. In many approaches, we have $n = m$, but some models use higher-dimensional relation embeddings. Embedding models define a scoring function $\phi : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, which indicates the plausiblity of the triple (e, r, f). Specifically, models are trained such that $\phi(\mathbf{e}, \mathbf{r}, \mathbf{f})$ is higher for triples (e, r, f) that appear in a given KG than for triples which do not.

The scoring function of many popular shallow models can expressed as $\phi(\mathbf{e}, \mathbf{r}, \mathbf{f}) = h(g_{\mathbf{r}}(\mathbf{e}), \mathbf{f})$ where $g_{\mathbf{r}}$ is a relation-specific transformation of $\mathbf{e}$ and h is a comparison function between two vectors. The main comparison functions for shallow models are of the following form:

$$h_1(g_{\mathbf{r}}(\mathbf{e}), \mathbf{f}) = -d(g_{\mathbf{r}}(\mathbf{e}), \mathbf{f}) + \lambda$$
$$h_2(g_{\mathbf{r}}(\mathbf{e}), \mathbf{f}) = g_{\mathbf{r}}(\mathbf{e}) \cdot \mathbf{f}$$

where $d : \mathbb{R}^n \times \mathbb{R}^n \to [0, +\infty)$ is a dissimilarity function, which is often taken to be the Euclidean distance or the squared Euclidian distance, and

$\lambda > 0$ is a bias term. Models of this form are, respectively, referred to as *linear* and *bilinear* models. TransE (Bordes et al., 2013) is a prominent example of a linear model, with $g_{\mathbf{r}}(\mathbf{e}) = \mathbf{e}+\mathbf{r}$. DistMult (Yang et al., 2015) is a prominent example of a bilinear model, with $g_{\mathbf{r}}(\mathbf{e}) = \mathbf{e} \odot \mathbf{r}$, where we write $\odot$ for the component-wise (i.e. Hadamard) product. The bias term λ was not included in the original formulation of TransE but it plays an important role in recent implementations of linear models (Sun et al., 2019), as it enables training such models using binary cross-entropy. We then interpret $\sigma(\phi(\mathbf{e}, \mathbf{r}, \mathbf{f}))$ as the probability that the triple (e, r, f) is valid, where we write σ for the sigmoid function. Without the bias term, for linear models we would always have $\phi(\mathbf{e}, \mathbf{r}, \mathbf{f}) \leq 0$ and thus $\sigma(\phi(\mathbf{e}, \mathbf{r}, \mathbf{f})) \leq 0.5$.

The comparison functions h_1 and h_2 are in fact closely related. Indeed, we have:

$$2 \times h_2(g_{\mathbf{r}}(\mathbf{e}), \mathbf{f}) = -d(g_{\mathbf{r}}(\mathbf{e}), \mathbf{f}) + b_{e,r,f} \quad (1)$$

where d is $\|\cdot\|^2$ and $b_{e,r,f} = \|g_{\mathbf{r}}(\mathbf{e})\|^2 + \|\mathbf{f}\|^2$. The main difference between h_1 and h_2 is then that the bias term of h_1 is constant, whereas the bias term of h_2 depends on the scored triple.

MuRE MuRE (Balazevic et al., 2019a) bridges the gap between linear and bilinear models by using a scoring function of the following form:

$$-\|(\mathbf{e} \odot \mathbf{s}+\mathbf{u}) - \mathbf{f}\|^2 + b_e + b_f \quad (2)$$

where relation vectors consist of two components: a scaling vector $\mathbf{s}$ and a translation $\mathbf{u}$; we have $\mathbf{e}, \mathbf{f}, \mathbf{s}, \mathbf{u} \in \mathbb{R}^n$, and $b_e, b_f \in \mathbb{R}$ are entity-specific biases. MuRE generalizes both TransE and DistMult. We recover TransE, for the case where d is the squared Euclidean distance, if $\mathbf{s} = (1, ..., 1)$ and $b_e = b_f = \frac{\lambda}{2}$ are fixed. We also recover the DistMult scoring function, up to a constant factor 2, by fixing $\mathbf{u} = (0, ..., 0)$, except that the bias term $\|\mathbf{e} \odot \mathbf{s}\|^2$ depends on both the entity e and the relation r, whereas b_e only depends on e.

The question of how bias terms should be defined has not yet received much attention. The entity-specific biases used by MuRE make it possible to encode a statistical prior. Entities that often appear as the tail entity in a triple are then more likely to be predicted than entities which only rarely appear in such positions. However, as we will see in the experiments, the impact of these entity-specific biases tends to be small in practice.

Region-Based Embeddings Region-based embeddings (Gutiérrez-Basulto and Schockaert, 2018; Abboud et al., 2020; Pavlovic and Sallinger, 2023; Charpenay and Schockaert, 2024) are a family of KG embedding models which have been designed such that rules can be captured in a principled way. In such models, relations are modelled as geometric regions (although these regions are still parameterized using vectors). Let $X_r \subseteq \mathbb{R}^{2n}$ be a region representing the relation r, with n the dimensionality of the entity embeddings. We then say that the triple (e, r, f) is captured iff $\mathbf{e};\mathbf{f} \in X_r$, where we write ; for vector concatenation. The fact that a strict criterion exists to determine whether a triple is captured or not makes it possible to formally study which kinds of rules can be modelled, which we will come back to in the next section.

Region-based approaches differ in how the region X_r are defined. For instance, in the case of ExpressivE (Pavlovic and Sallinger, 2023), X_r is defined in terms of two-dimensional parallelograms. Specifically, relation r is then defined using parallelograms $X_1^r, ..., X_n^r$ and we say that (e, r, f) is captured if $(e_1, f_1) \in X_1^r, ..., (e_n, f_n) \in X_n^r$, where we assume $\mathbf{e} = (e_1, ..., e_n)$ and $\mathbf{f} = (f_1, ..., f_n)$. Charpenay and Schockaert (2024) proposed a model of the same form, where octagons are used instead of parallelograms. We can also consider region-based variants of standard shallow embedding models, such as MuRE. We can e.g. use regions of the following form:

$$X_i^r = \{(x, y) \mid u^- \leq \lambda y - x \leq u^+\} \quad (3)$$

Note how this corresponds to adding a learnable width parameter ($w = u^+ - u^-$) to MuRE. Region-based formulations of standard models are commonly obtained by simply changing the dissimilarity function to a piecewise-linear transformation of Euclidean distance $d_{\mathbf{w}}$, which incorporates this width. For instance, the scoring function of ExpressivE has the following form:

$$-d_{\mathbf{w}}([\mathbf{e}; \mathbf{f}] \odot \mathbf{s} + \mathbf{u}, [\mathbf{f}; \mathbf{e}]) + \lambda$$

Note that ExpressivE paralellograms are obtained via the same components as MuRE, i.e. a scaling vector $\mathbf{s}$ and a translation $\mathbf{u}$. If we set $\mathbf{w} = (0, \ldots, 0)$, we obtain a *functional* variant of the region-based model, with empty volume.

Cross-coordinate Comparisons Standard linear and bilinear models compare entity embeddings

coordinate-wise. To compute a score for entity embeddings $\mathbf{e} = (e_1, ..., e_n)$ and $\mathbf{f} = (f_1, ..., f_n)$, these models essentially compare each coordinate e_i with the corresponding coordinate f_i. The same is also true for the aforementioned region-based embeddings, which are defined in terms of two-dimensional regions. However, some models instead compare pairs of coordinates (e_i, e_j) with pairs of coordinates (f_j, f_i). This is the case for ComplEx (Trouillon et al., 2017), which uses a scoring function of the following form:

$$h_2([\mathbf{e};\mathbf{e}';\mathbf{e}';\mathbf{e}'] \odot [\mathbf{s};\mathbf{s}';\mathbf{s};-\mathbf{s}'], [\mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}]) \quad (4)$$

Note that the embedding of the entity e consists of two blocks, $\mathbf{e}$ and $\mathbf{e}'$, and similar for the embedding of the entity f and for the relation embedding. The name ComplEx refers to the fact that these two blocks can be interpreted as the real and imaginary parts of a vector of complex numbers. Note that the ComplEx scoring function involves comparing $\mathbf{e}$ with $\mathbf{f}$ and $\mathbf{e}'$ with $\mathbf{f}'$, which we refer to as coordinate-wise comparisons, as well as comparing $\mathbf{e}$ with $\mathbf{f}'$ and $\mathbf{e}'$ with $\mathbf{f}$, which we refer to as cross-coordinate comparisons. SimplE (Kazemi and Poole, 2018) is a special case of ComplEx which only involves cross-coordinate comparisons.

BoxE (Abboud et al., 2020) also involves cross-coordinate comparisons only, using a scoring function of the following form:

$$-d_{\mathbf{w}}([\mathbf{e};\mathbf{e}'] + \mathbf{u}, [\mathbf{f}';\mathbf{f}]) + \lambda$$

where $d_{\mathbf{w}}$ is a piecewise linear transformation of the Euclidean distance, in line with BoxE's formulation as a region-based model.

A Unified View Based on the previous discussion, we consider the following scoring function:

$$-d([\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}'] \odot \mathbf{s} + \mathbf{u}, [\mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}]) + b_{e,r,f}$$

where d is either the squared Euclidean distance, or a transformation of the Euclidean distance, and $b_{e,r,f} > 0$ is a trainable bias, which in general may be entity-specific and/or relation-specific. We have $[\mathbf{e};\mathbf{e}'], [\mathbf{f};\mathbf{f}'] \in \mathbb{R}^n$ and $\mathbf{s}, \mathbf{u} \in \mathbb{R}^{2n}$. As special cases, we can obtain variants of most of the aforementioned models. Indeed, TransE and DistMult correspond to variants where the terms with cross-coordinate comparisons are omitted, where the (squared) Euclidean distance is used as the dissimilarity function, and where respectively $\mathbf{s} = (1, ..., 1)$ and $\mathbf{u} = (0, ..., 0)$ are fixed. ComplEx corresponds to a variant where $\mathbf{u} = (0, ..., 0)$ is fixed, $\mathbf{s}$ is of the form $[\mathbf{s_1};\mathbf{s_2};\mathbf{s_1};-\mathbf{s_2}]$ and d is the squared Euclidean distance. BoxE corresponds to a variant in which the coordinate-wise comparisons are omitted and $\mathbf{s} = (1, ..., 1)$. The region-based variant of MuRE corresponds to the case where the cross-coordinate comparisons are dropped and d is chosen as $d_{\mathbf{w}}$.

The region-based variant of MuRE is similar to ExpressivE, if inverse relations are added to the KG, since parallelograms correspond to the intersection of two MuRE regions of the form (3). This data augmentation technique, first introduced by Lacroix et al. (2018), is commonly used when learning KG embeddings. However, we will show in the experiments that ExpressivE, which can be seen as a straightforward extension of MuRE, outperforms MuRE and all other evaluated models. Notably, ExpressivE applies a transformation to both $\mathbf{e}$ and $\mathbf{f}$, whereas $g_{\mathbf{r}}$ applies only to $\mathbf{e}$ in our formulation, which gives it more flexibility for modeling N-N relations.

4 Expressivity Results

Region-based models allow us to study precisely what kind of knowledge a given model can capture. In this section, we study this for the region-based formulation of MuRE, which was introduced in (3) and which we now formally define.

Definition 1 (MuRE region embedding)**.** *An (n-dimensional) MuRE region embedding is a triple (Z, γ, τ), with $Z \subseteq \mathbb{R}^n$, $\gamma : \mathcal{E} \to Z$ and $\tau : \mathcal{R} \to \mathbb{R}^2 \times ... \times \mathbb{R}^2$, such that for each $r \in \mathcal{R}$, we have $\tau(r) = (X_1^r, ..., X_n^r)$ with X_i^r a two-dimensional region of the form (3). We call γ a (MuRE) entity embedding and τ a (MuRE) relation embedding.*

In this definition, we assume that all entity embeddings belong to the set Z. In previous work (Abboud et al., 2020; Pavlovic and Sallinger, 2023; Charpenay and Schockaert, 2024), $Z = \mathbb{R}^n$ is implicitly assumed. However, as will become clear, the ability to restrict Z to a subset of $\mathbb{R}^n$ matters in the case of MuRE regions. When Z is clear from the context, we will usually write MuRE embeddings as pairs (γ, τ). We write $\gamma_i(e)$ for the i^{th} coordinate of $\gamma(e)$, and $\tau_i(r)$ for the corresponding two-dimensional region, i.e. we we have $\gamma(e) = (\gamma_1(e), ..., \gamma_n(e))$ and $\tau(r) = (\tau_1(r), ..., \tau_n(r))$. For $\mathbf{e} = (e_1, ..., e_n)$, $\mathbf{f} = (f_1, ..., f_n)$, $r \in \mathcal{R}$, and τ a relation embedding, we write $\tau \models r(\mathbf{e}, \mathbf{f})$ to

denote that $(e_i, f_i) \in \tau_i(r)$ for each $i \in \{1, ..., n\}$.

Definition 2 (Capturing triples)**.** *We say that a triple $(e, r, f) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is captured by the embedding (γ, τ) iff $\tau \models r(\gamma(e), \gamma(f))$. In this case, we write $(\gamma, \tau) \models r(e, f)$.*

We first show that MuRE regions can capture any KG over $\mathcal{E}$ and $\mathcal{R}$.

Proposition 1 (Full expressivity)**.** *Let $\mathcal{G} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. There exists a MuRE region embedding (γ, τ) such that $(\gamma, \tau) \models r(e, f)$ iff $(e, r, f) \in \mathcal{G}$.*

We also want KG embeddings to capture rules, to ensure that relation embeddings reflect the regularities that exist in the domain. Given a closed path rule $r_1(X_1, X_2) \wedge ... \wedge r_p(X_p, X_{p+1}) \rightarrow s(X_1, X_{p+1})$, we intuitively want to ensure that whenever the triples $(e_1, r_1, e_2), ..., (e_p, r_p, e_{p+1})$ are captured by the embedding, for some entities $e_1, ..., e_{p+1}$, then the triple (e_1, s, e_{p+1}) is also captured. However, Gutiérrez-Basulto and Schockaert (2018) proposed a more general definition, which does not refer to particular entity embeddings. This is because we want the relation embeddings to guarantee that the rule remains satisfied, even if entities are later added to the KG, which is important in the setting of inductive KG completion (Teru et al., 2020). This is made precise as follows.

Definition 3 (Capturing rules)**.** *Let $Z \subseteq \mathbb{R}^n$ be the considered domain of the entity embeddings. We say that a closed path rule $r_1(X_1, X_2) \wedge ... \wedge r_p(X_p, X_{p+1}) \rightarrow s(X_1, X_{p+1})$ is captured by a relation embedding τ iff for all $\mathbf{e_1}, ...\mathbf{e_{p+1}} \in Z$ such that $\tau \models r_1(\mathbf{e_1}, \mathbf{e_2}), ..., \tau \models r_p(\mathbf{e_p}, \mathbf{e_{p+1}})$, it also holds that $\tau \models s(\mathbf{e_1}, \mathbf{e_{p+1}})$.*

We write $\tau \models \rho$ to denote that τ captures the closed path rule ρ. Ideally, we would want a region-based embedding model to be such that for every set of closed path rules $\mathcal{K}$, we can find a relation embedding that captures the rules that are entailed by $\mathcal{K}$, and only those rules. Unfortunately, this is not possible for existing region based models. However, Charpenay and Schockaert (2024) showed that this property can be satisfied by their octagon embeddings, as long as every rule that is entailed by $\mathcal{K}$ satisfies the following condition of regularity (except for trivial rules of the form $r(X, Y) \subseteq r(X, Y)$).

Definition 4. *A closed path rule $r_1(X_1, X_2) \wedge ... \wedge r_p(X_p, X_{p+1}) \rightarrow s(X_1, X_{p+1})$ is called* regular *if $r_1, ...r_p, s$ are all distinct relations.*

We may wonder whether, similar to octagon embeddings, MuRE region embeddings can also capture sets of regular rules. Unfortunately, for $Z = \mathbb{R}^n$, this is not the case (as we formally show in Appendix A.3). However, if we assume that all entity embeddings occur in a bounded region, then MuRE regions can capture sets of regular rules.

Proposition 2. *Let $Z = [z_1^-, z_1^+] \times ... \times [z_n^-, z_n^+]$, with $z_i^- < z_i^+$ for all $i \in \{1, ..., n\}$. Let $\mathcal{K}$ be a set of closed path rules. Assume that any closed path rule entailed by $\mathcal{K}$ (in terms of classical logic entailment) is either a trivial rule of the form $r(X, Y) \rightarrow r(X, Y)$ or a regular rule. There exists a relation embedding τ which captures all rules entailed by $\mathcal{K}$, and only those rules.*

Thus far, octagon embeddings were the only coordinate-wise region-based embeddings known to be capable of capturing sets of regular rules. The fact that MuRE regions can do the same is remarkable, given that these regions are much simpler, being defined by 3 parameters rather than 8. In fact, the proof of Proposition 2 involves a restricted family of MuRE regions, which are characterized by only 2 parameters. Existing embedding models (Abboud et al., 2020; Pavlovic and Sallinger, 2023; Charpenay and Schockaert, 2024) are also capable of capturing the following types of rules:

Hierarchy: $r_1(X, Y) \rightarrow r_2(X, Y)$

Intersection: $r_1(X, Y) \wedge r_2(X, Y) \rightarrow r_3(X, Y)$

Symmetry: $r_1(X, Y) \rightarrow r_1(Y, X)$

Inversion: $r_1(X, Y) \rightarrow r_2(Y, X)$

where the notion of capturing such rules is defined entirely analogously as for closed path rules. MuRE regions can also capture arbitrary sets of such rules, even for $Z = \mathbb{R}^n$, and even when all scale factors are 1. Specifically, let us refer to a MuRE region $X = \{(x, y) \mid u^- \leq \lambda y - x \leq u^+\}$ as a TransE region if $\lambda = 1$.

Proposition 3. *Let $Z = [z_1^-, z_1^+] \times ... \times [z_n^-, z_n^+]$, with $z_i^- < z_i^+$ for all $i \in \{1, ..., n\}$, or $Z = \mathbb{R}^n$. Let $\mathcal{K}$ be a set of hierarchy, intersection, symmetry and inversion rules. There exists a relation embedding τ which captures all rules entailed by $\mathcal{K}$, and only those rules, such that $\tau_i(r)$ is a TransE region for every $r \in \mathcal{R}$ and $i \in \{1, ..., n\}$.*

However, MuRE regions are limited when it comes to combining symmetry rules and closed path rules. Transitivity rules of the form $r(X, Y) \wedge r(Y, Z) \rightarrow r(X, Z)$ also play an important role in many domains. However, such rules can only be captured in an approximate way when using MuRE regions.

Further discussion about these limitations is presented in Appendix A.6.

5 Experiments and Analysis

In Section 3, we identified five orthogonal design choices for defining shallow KG embedding models: (i) the choice of the dissimilarity function d; (ii) whether and which kinds of trainable biases are used; (iii) whether the head entity is transformed using scaling, translation, or both; (iv) whether cross-coordinate comparisons are used; and (v) whether a region-based formulation with a trainable width is used. An exhaustive evaluation of all combinations on benchmarks such as WN18RR and FB15k-237 is not feasible. We therefore first carry out an extensive evaluation on three popular small-scale datasets (Countries, Kinships and UMLS), which we complement with an evaluation on synthetic datasets. We then evaluate the best performing variants on the larger WN18RR and FB15k-237 datasets. We report the most common metrics for link prediction: Hits@k and Mean Reciprocal Rank (MRR). All models were implemented in PyKEEN (Ali et al., 2021). Our implementation, along with instructions for reproducibility, is available online.[1]

5.1 Small-scale Knowledge Graphs

Countries is a KG about countries and continents with only two relations: *is-inside*, relating countries with continents, and *is-neighbour*, relating countries with each other (Bouchard et al., 2015). This is an artificial and simple KG, used to evaluate whether models can capture symmetry and closed-path rules. *Kinships* and *UMLS*, on the other hand, have been used early on as real-world examples of multi-relational datasets (Bordes et al., 2014). *Kinships* features 26 kinship relations (as defined by the Australian Alyawarra tribe). The dataset is challenging due to the complex nature of the kinship relationships that are considered, and due to the presence of some noise. *UMLS* is an excerpt of the Unified Medical Language System, defining relationships between diseases, symptoms and body parts, among others (49 relations). All three datasets have between 100 and 300 entities. In our experiments, we fix entity embeddings to be 40-dimensional. For models with entity embeddings of the form $[\mathbf{e};\mathbf{e}']$, such as ComplEx and BoxE, we set $\mathbf{e} \in \mathbb{R}^{20}$ and $\mathbf{e}' \in \mathbb{R}^{20}$. Other hyperparameters are also fixed. Results are summarised in Table 1. The table is divided in five blocks, corresponding to the five considered design choices.

Results in the first and second blocks of Table 1 suggest that linear models, which score triples with a norm instead of a dot product, perform better than bilinear models. Intuitively, linear models introduce a bias term that is not dependent on entity embeddings, giving more degrees of freedom. Making this bias term trainable further improves the performance of linear models, as already observed by Balazevic et al. (2019a). We extend their observation by considering relation-specific biases (b_r) in addition to entity-specific terms ($b_e + b_f$). We found that entity embeddings can capture a statistical prior for each entity, correlating with the degree of entities in the KG. Such an encoding does not help to predict links if the KG is balanced (as in Kinships, where the correlation between entity biases and degrees is close to 0) or if entities are highly connected (as in UMLS). More details can be found in Appendix B.5.

Results in the third block indicate that combining scaling and translation is important. Translation-only variants (featuring $g_{\mathbf{1};\mathbf{u}}$) have poor results. It is worth noting that some models in the literature also allow transformation through rotation[2]. RESCAL, for instance, embeds relations as arbitrary linear transformations, combining rotation with scaling (Nickel et al., 2011b). Later models extended it to affine transformations, also featuring translation (Jiang et al., 2024; Ge et al., 2022). Despite a significantly higher number of trainable parameters, the performance of these models does not exceed that of MuRE or RotatE, suggesting that scaling and translation are sufficient.

Cross-coordinate comparisons tend to decrease the performance of MuRE variants, as can been seen in the fourth block. It only benefits models on Kinships, a KG with many symmetries (where limitations of MuRE discussed in Appendix A.6 arise). We further explore the combination of symmetry and closed-path rules in the next section. Combining coordinate-wise and cross-coordinate comparison, as in ComplEx, does not improve performance either[3]. The last block shows results

[1] https://github.com/vcharpenay/shallow-kges

[2] We refer here to rotation in $\mathbb{R}^n$. Note that RotatE, despite its name, should rather be considered as a scaling model: rotation is not in $\mathbb{R}^n$ but in the two-dimensional Euler plane.

[3] The poor performance of ComplEx on Countries is likely due to its regularization scheme, which is ineffective on small datasets

	Countries				Kinships				UMLS			
	h@1	h@3	h@10	MRR	h@1	h@3	h@10	MRR	h@1	h@3	h@10	MRR
$g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) \cdot \mathbf{f}$	.404	.762	.945	.602	.418	.696	.940	.588	.674	.935	.983	.808
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e})-\mathbf{f}\|^2 + \|g_{\mathbf{s};\mathbf{u}}(\mathbf{e})\|^2 + \|\mathbf{f}\|^2$	.341	.641	.837	.517	.434	.702	.939	.600	.703	.932	.986	.823
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e})-\mathbf{f}\|^2 + \lambda$	.504	.870	.945	.688	.476	.741	.943	.632	.760	.969	.994	.865
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e})-\mathbf{f}\| + \lambda$	.608	.962	**1.0**	.790	.492	.745	.937	.643	.767	.962	.989	.867
MuRE (2)	.675	.979	**1.0**	.820	.478	.736	.945	.633	.775	**.977**	**.996**	.877
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e})-\mathbf{f}\| + b_e + b_f$	.720	.891	.937	.809	.490	.748	.944	.643	.764	.967	.994	.867
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e})-\mathbf{f}\| + b_r$	.683	**.995**	.995	.828	.496	.745	.939	.645	.772	.971	.993	.873
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + \lambda$ (TransE)	.262	.441	.520	.367	.026	.069	.190	.089	.475	.650	.798	.588
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e})-\mathbf{f}\| + b_r$	.695	.954	.983	.825	.467	.722	.928	.621	.742	.940	.984	.846
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e})-\mathbf{f}\| + b_r$	.729	.979	**1.0**	.853	.071	.146	.304	.151	.479	.812	.943	.660
$g_{\mathbf{s};\mathbf{0}}([\mathbf{e};\mathbf{e}']) \cdot [\mathbf{f}';\mathbf{f}]$ (SimplE)	.104	.275	.504	.234	.343	.582	.875	.506	.555	.722	.883	.664
ComplEx (4)	0	.008	.024	.016	.633	.857	.971	.757	.774	.949	.985	.865
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}')-\mathbf{f}';\mathbf{f}\| + b_r$	.650	.983	.991	.809	.633	.861	.971	.757	.763	.961	.988	.865
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}')-\mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	.620	.937	**1.0**	.778	**.658**	**.880**	**.978**	**.778**	.736	.936	.985	.841
$-d_{\mathbf{w}}(g_{\mathbf{1};\mathbf{u}}(\mathbf{e};\mathbf{e}')-\mathbf{f}';\mathbf{f}) + \lambda$ (BoxE)	**.804**	.945	**1.0**	**.880**	.517	.782	.957	.669	.766	.958	.984	.865
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{f})-\mathbf{f};\mathbf{e}) + \lambda$ (ExpressivE)	.466	.712	.850	.609	.584	.821	.965	.718	**.816**	.970	.993	**.895**
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e})-\mathbf{f}) + b_r$	.495	.770	.916	.644	.520	.784	.959	.673	.767	.957	.985	.863
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}')-\mathbf{f}';\mathbf{f}) + b_r$	.395	.616	.800	.537	.539	.798	.963	.686	.739	.950	.983	.848

Table 1: Link prediction performance on small-scale datasets, where $g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) = \mathbf{e} \odot \mathbf{s} + \mathbf{u}$, λ is a margin (fixed bias), b_e, b_f, b_r are trainable biases specific to an entity or a relation, $\mathbf{x};\mathbf{y}$ is the concatenation of $\mathbf{x}$ and $\mathbf{y}$, and $d_{\mathbf{w}}$ is the piecewise linear function of BoxE (results are averaged across 5 runs).

	CPR				CPR-rt				CPR-rtu			
	h@1	h@3	h@10	MRR	h@1	h@3	h@10	MRR	h@1	h@3	h@10	MRR
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	.015	.121	**.656**	.164	.002	.053	.520	.121	.003	.354	.571	.214
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	.252	.432	.613	.375	**.312**	**.624**	**.871**	**.501**	.494	.661	.786	.598
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	.111	.173	.271	.163	.154	.274	.440	.248	**.529**	**.701**	**.834**	**.634**
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	.115	.175	.247	.163	.150	.249	.394	.231	.506	.679	.803	.610
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	**.309**	**.450**	.648	**.417**	.294	.519	.763	.445	.462	.589	.711	.548

Table 2: Link prediction performance on small-scale synthetic datasets.

for region-based models, which introduce a width parameter ($\mathbf{w}$). Interestingly, the interaction of cross-coordinate comparison and trainable widths in BoxE has a positive effect. Overall, MuRE variants are outperformed by ExpressivE, a region-based extension of MuRE. Further experiments in Appendix B.3 show that even functional ExpressivE ($\mathbf{w} = \mathbf{0}$) outperforms MuRE on all datasets.

5.2 Synthetic Knowledge Graphs

When comparing coordinate-wise with cross-coordinate approaches, the results so far are mixed: models with cross-coordinate comparison have a clear advantage on Kinships but they underperform on Countries. As already mentioned in Section 4, MuRE is limited when it comes to learning rule bases that combine symmetry and closed-path rules. In the following, we complement our findings with experiments on synthetic KGs. We generated three datasets, as follows. For the first dataset, we randomly sample triples of the form (a, r, b) and (a, s, b), and subsequently apply the rule $r(X, Y) \wedge s(Y, Z) \rightarrow t(X, Z)$ to add triples of the form (a, t, c). Half of the generated t-triples are shown during training, the other half is kept for evaluation. Note that test triples can thus be inferred from the given KG. We refer to this dataset as CPR. The second dataset has the same triples but r and t are made symmetric. In other words, we add all triples that can be inferred using the rules $r(X, Y) \rightarrow r(Y, X)$ and $t(X, Y) \rightarrow t(Y, X)$. The test set consists of half of the t-triples, as in the first dataset. We refer to this dataset as CPR-rt. The last dataset is obtained by randomly sampling triples $r(a, b), s(a, b), t(a, b)$ and subsequently applying the rule $r(X, Y_1) \wedge s(Y_1, Y_2) \wedge t(Y_2, Z) \rightarrow u(X, Z)$, and making r, t and u symmetric. On this dataset, half of the u-triples are used for evaluation. We refer to this last dataset as CPR-rtu.

The results in Table 2 show that cross-coordinate

	WN18RR				FB15k-237			
	h@1	h@3	h@10	MRR	h@1	h@3	h@10	MRR
Functional ExpressivE	.438	**.503**	**.580**	**.486**	**.219**	**.355**	**.515**	**.318**
ExpressivE	.430	.475	.534	.464	.185	.311	.471	.279
BoxE	.389	.429	.464	.417	.197	.316	.476	.288
RotatE	**.444**	.495	.552	.480	.216	.345	.502	.311
ComplEx	.391	.409	.431	.405	.130	.217	.338	.199
QuatE	.426	.464	.502	.452	.159	.278	.438	.250
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	.400	.455	.497	.436	.204	.326	.485	.295
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$ (w/ inverse)	.427	.484	.553	.469	.212	.338	.503	.307
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	.325	.391	.439	.367	.200	.318	.473	.289
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	.361	.422	.460	.399	.201	.324	.483	.293
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}) + b_r$	.264	.304	.345	.292	.183	.286	.432	.264

Table 3: Link prediction performance on benchmark datasets.

comparison models outperform coordinate-wise models on CPR-rtu. On the other datasets, however, cross-coordinate comparison models underperform. Furthermore, MuRE captures closed-path rules more efficiently than the translation-only model. The model with both coordinate-wise and cross-coordinate comparison performs well on average. Interestingly, on CPR-rtu the cross-coordinate variant without translation performs better than the one with both scaling and translation. More results can be found in Appendix B.6.

5.3 Benchmark Knowledge Graphs

We now analyze the performance of MuRE variants on WN18RR and FB15k-237. So far, we found that linear models with trainable biases generally outperform, and that scaling and translation are both useful. We selected four of the previously considered variants, all with these properties, to compare with state-of-the-art shallow models. As experiments on small KGs showed that (functional) ExpressivE outperforms MuRE, we also included a fifth configuration, obtained by systematically adding inverse relations to the KG. This new configuration based on the coordinate-wise MuRE variant is closest to functional ExpressivE.

5.3.1 Main Experiments

Results for existing models are available in the literature, but they depend on training configurations that are not always comparable. For a fair comparison, we ran our own experiments for all models, ensuring that models are trained in the same setting for a given dataset. We discuss this aspect in further detail in Appendix B.7. We chose $n = 100$ for WN18RR and $n = 200$ for FB15k-237. We use a binary cross-entropy loss with self-adversarial negative sampling (Sun et al., 2019) for all configurations.

From previous experiments, we found that the impact of cross-coordinate comparisons and trainable widths is mixed, partly influenced by symmetries. These observations generalize to WN18RR and FB15k-237. The variants with cross-coordinate comparisons and trainable widths each perform poorly but BoxE, which combines the two features, outperforms them. Interestingly, QuatE performs significantly better than ComplEx in our setting. Both models are bilinear but QuatE introduces more cross-coordinate comparisons than ComplEx, splitting embeddings in four blocks instead of two. A linear variant of QuatE may compete with the best-performing models but QuatE itself underperforms MuRE.

Functional ExpressivE is the best performing model, followed by RotatE. The performance of the baseline MuRE variant significantly improves if inverse relations are added to the KG, matching RotatE on hits@10. We conjecture that functional ExpressivE performs better than MuRE, despite the fact that the two models have the same number of parameters, because it can better handle N-N relations. It takes twice as long to train ExpressivE, however (see training times in Appendix B.7).

5.3.2 Impact of Negative Sampling

The results we obtained for ComplEx are significantly below the best published results for this model. For instance, in our setting, ComplEx achieves an MRR of .199 on FB15k-237 while Lacroix et al. (2018) report .35 for the same value of n. This gap can be explained as follows. In their setting, called 1-vs-all, ComplEx has been trained with significantly more compute. The 1-vs-all setting involves comparing every triple $r(e, f)$ in the graph to $|\mathcal{E}| - 1$ other triples $r(e, f')$, where

	WN18RR			
	h@1	h@3	h@10	MRR
ComplEx	.43	.47	.52	.46
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	**.45**	**.50**	**.56**	**.49**
	FB15k-237			
	h@1	h@3	h@10	MRR
ComplEx	**.26**	**.39**	**.54**	**.35**
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	.22	.35	.51	.32

Table 4: Link prediction performance on benchmark datasets in the 1-vs-all setting; other hyperparameters were left unchanged; results for ComplEx by Lacroix et al. (2018).

$|E| \sim 15,000$ in FB15k-237. In contrast, we only compare triples to 50 randomly sampled triples (see the hyperparameters in Table 5). The average out-degree in FB15k-237 being around 3, 1-vs-all training thus requires scoring 100 times more triples than our training setting ($15,000$ vs. 3×50). We chose the more efficient setting for our main experiments because computational efficiency is an important quality of shallow KGE models. However, given the disappointing results for ComplEx, we now complement our main results with an analysis in the 1-vs-all setting.

Table 4 compares the performance of the baseline MuRE model with ComplEx in the 1-vs-all setting. The results show that MuRE also benefits from more compute, but not equally across datasets: it significantly outperforms ComplEx on WN18RR, which is consistent with our observations in the main experiments, but it underperforms ComplEx on FB15k-237. This additional experiment highlights that a trade-off is to be found between performance and computational efficiency. If both are important in downstream tasks, MuRE remains a good baseline. In the literature, RotatE, BoxE, ExpressivE and QuatE were all trained with negative sampling. We therefore limit our comparison to ComplEx in this experiment.

6 Conclusions

We have systematically analyzed shallow embedding models, taking MuRE as our base model. Among others, we found that linear models generally outperform their bilinear counterparts. We also found that including cross-coordinate comparisons is important on some datasets, but not all. Finally, we theoretically showed that MuRE is surprisingly expressive, matching what is known for state-of-the-art region based models. We therefore recommend to consider MuRE and its extension, functional ExpressivE, as the new baselines for shallow KG embeddings.

Acknowledgments

Computations have been performed on the supercomputer facilities of the Mésocentre Clermont-Auvergne of the Université Clermont Auvergne. Steven Schockaert was supported EPSRC Grant EP/W003309/1.

Limitations

Our theoretical espressivity results only establish lower bounds, i.e. we have provided examples of types of rule bases that MuRE regions can capture, but there may be other types of rule bases that can be captured as well. The same is true for what is known about existing region-based models. As such, while we have established that MuRE regions can match what is known about existing region based models, in terms of expressivity, there might still exist types of rule bases that can be captured using, for instance, octagons or parallelograms, but not using MuRE regions. There might also be a difference between what can be captured in theory, and which kinds of rules can be effectively learned in practice. For instance, the construction which shows that MuRE regions can capture sets of regular closed-path rules involves very small constants, so the representations from that particular construction would be difficult to learn in practice.

In our empirical analysis, we have analyzed the impact of different scoring functions, but the training configuration was kept fixed. It is possible that different conclusions might be reached, for instance, if some models are trained with lower learning rates or different regularizers. Our analysis is also limited by the specific choice of datasets that were considered. While the impact of some design choices impacts the performance consistently across all datasets, for other design choices, the effects are dataset-specific, and further analysis is needed to characterize more precisely when certain design choices are useful.

References

Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. BoxE: A box embedding model for knowledge base completion. In *Advances in Neural Information*

Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Mikhail Galkin, Sahand Sharifzadeh, Asja Fischer, Volker Tresp, and Jens Lehmann. 2022. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):8825–8845.

Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. 2021. PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings. *Journal of Machine Learning Research*, 22(82):1–6.

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019a. Multi-relational poincaré graph embeddings. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4465–4475.

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019b. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5184–5193. Association for Computational Linguistics.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation. *Mach. Learn.*, 94(2):233–259.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Guillaume Bouchard, Sameer Singh, and Théo Trouillon. 2015. On approximate reasoning capabilities of low-rank vector spaces. In *2015 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 22-25, 2015*. AAAI Press.

Jiahang Cao, Jinyuan Fang, Zaiqiao Meng, and Shangsong Liang. 2024. Knowledge graph embedding: A survey from the perspective of representation spaces. *ACM Comput. Surv.*, 56(6):159:1–159:42.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6901–6914. Association for Computational Linguistics.

Victor Charpenay and Steven Schockaert. 2024. Capturing knowledge graphs and rules with octagon embeddings. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 3289–3297. ijcai.org.

Nguyen Nam Doan, Aki Härmä, Remzi Celebi, and Valeria Gottardo. 2024. A hybrid retrieval approach for advancing retrieval-augmented generation systems. In *Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)*, pages 397–409, Trento. Association for Computational Linguistics.

Xiou Ge, Yun-Cheng Wang, Bin Wang, and C. C. Jay Kuo. 2022. Compounde: Knowledge graph embedding with translation, rotation and scaling compound operations. *Preprint*, arXiv:2207.05324.

Víctor Gutiérrez-Basulto and Steven Schockaert. 2018. From knowledge graph embedding to ontology embedding? an analysis of the compatibility between vector space representations and rules. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018*, pages 379–388. AAAI Press.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. *Knowledge Graphs*. Number 22 in Synthesis Lectures on Data, Semantics, and Knowledge. Springer.

Wenyu Huang, Guancheng Zhou, Mirella Lapata, Pavlos Vougiouklis, Sébastien Montella, and Jeff Z. Pan. 2024. Prompting large language models with knowledge graphs for question answering involving long-tail facts. *CoRR*, abs/2405.06524.

Jiahao Jiang, Fei Pu, Jie Cui, and Bailin Yang. 2024. Affine transformation-based knowledge graph embedding. In *Knowledge Science, Engineering and Management*, pages 284–297, Singapore. Springer Nature Singapore.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. Knowledge base completion: Baselines strike back. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 69–74, Vancouver, Canada. Association for Computational Linguistics.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning, ICML*

2023, 23-29 July 2023, Honolulu, Hawaii, USA, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 4289–4300.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2869–2878. PMLR.

Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical inference for multi-relational embeddings. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2168–2178. PMLR.

Xin Lv, Yankai Lin, Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. Do pre-trained models benefit knowledge graph completion? A reliable evaluation and a reasonable approach. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3570–3581. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics.

Nicholas Matsumoto, Jay Moran, Hyunjun Choi, Miguel E Hernandez, Mythreye Venkatesan, Paul Wang, and Jason H Moore. 2024. Kragen: a knowledge graph-enhanced rag framework for biomedical problem solving using large language models. *Bioinformatics*, 40(6).

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016a. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. 2016b. Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 1955–1961. AAAI Press.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011a. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 809–816. Omnipress.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011b. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 809–816. Omnipress.

Aleksandar Pavlovic and Emanuel Sallinger. 2023. 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*. OpenReview.net.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2020. You CAN teach an old dog new tricks! on training knowledge graph embeddings. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. 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.

Komal K. Teru, Etienne G. Denis, and William L. Hamilton. 2020. Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9448–9457. PMLR.

Théo Trouillon, Christopher R. Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. 2017. Knowledge graph completion via complex tensor factorization. *J. Mach. Learn. Res.*, 18:130:1–130:38.

Théo Trouillon and Maximilian Nickel. 2017. Complex and holographic embeddings of knowledge graphs: A comparison. *CoRR*, abs/1707.01475.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 950–958. ACM.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Trans. Assoc. Comput. Linguistics*, 9:176–194.

Yanbin Wei, Qiushi Huang, Yu Zhang, and James T. Kwok. 2023. KICGPT: large language model with knowledge in context for knowledge graph completion. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 8667–8683. Association for Computational Linguistics.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Han Yang and Junfei Liu. 2021. Knowledge graph representation learning as groupoid: Unifying transe, rotate, quate, complex. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 2311–2320. ACM.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019a. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2731–2741.

Yichi Zhang, Zhuo Chen, Lingbing Guo, Yajing Xu, Wen Zhang, and Huajun Chen. 2024. Making large language models perform better in knowledge graph completion. In *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pages 233–242. ACM.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019b. ERNIE: enhanced language representation with informative entities. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1441–1451. Association for Computational Linguistics.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal A. C. Xhonneux, and Jian Tang. 2021. Neural bellman-ford networks: A general graph neural network framework for link prediction. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 29476–29490.

A Proof of Expressivity Results

In the following, we write $\mathsf{Reg}(u^-, u^+, \lambda)$ for a region of the form (3).

A.1 Proof of Proposition 1

The proofs follows the same strategy as the full expressivity proof for octagon embeddings (Charpenay and Schockaert, 2024). However, because MuRE regions are more constrained than octagons, we need to double the number of coordinates in the construction. In particular, we represent each entity e as a vector $\gamma(e) = \mathbf{e}$ of dimension $n = 2 \cdot |\mathcal{R}| \cdot |\mathcal{E}|$. In particular, for each relation $r \in \mathcal{R}$ and each entity $f \in \mathcal{E}$, we have two corresponding coordinates. We will write $e_{r,f,1}$ and $e_{r,f,2}$ for the value of these coordinates in the embedding of entity e. We show that there exists a MuRE region embedding which satisfies a triple (e, r, f) iff it belongs to $\mathcal{G}$. We choose the coordinates of $\mathbf{e}$ as follows:

$$e_{r,f,1} = e_{r,f,2} = \begin{cases} 0 & \text{if } e = f \\ 1 & \text{if } e \neq f \text{ and } (f, r, e) \in \mathcal{G} \\ 2 & \text{otherwise} \end{cases}$$

Let us write $M^s_{r,e}$ for the MuRE region representing relation s in the coordinate associated with entity e and relation r, and similar for $M^s_{r,1}$ and $M^s_{r,2}$. For all coordinates where $s \neq r$, we choose $M^s_{r,e,1} = M^s_{r,e,2} = \mathsf{Reg}(-2, 2, 1)$. The regions $M^r_{r,e,1}$ and $M^r_{r,e,2}$ are chosen as follows:

- If $(e, r, e) \in \mathcal{G}$ we choose $M^r_{r,e,1} = M^r_{r,e,2} = \mathsf{Reg}(-2, 1, 1)$.
- Otherwise, we choose $M^r_{r,e,1} = \mathsf{Reg}(-2, 1, 1)$ and $M^r_{r,e,2} = \mathsf{Reg}(-4, -1, -1)$.

In the following, what matters is which of the points (i, j), with $i, j \in \{0, 1, 2\}$, belong to the considered MuRE regions. We can make the following observations:

- $\mathsf{Reg}(-2, 2, 1)$ contains all points (i, j) with $i, j \in \{0, 1, 2\}$.
- $\mathsf{Reg}(-2, 1, 1)$ contains all points (i, j) with $i, j \in \{0, 1, 2\}$, apart from $(0, 2)$.
- $\mathsf{Reg}(-4, -1, -1)$ contains all points (i, j) with $i, j \in \{0, 1, 2\}$, apart from $(0, 0)$.

Suppose $(e, r, f) \in \mathcal{G}$. We need to show that $(e_{s,g}, f_{s,g}) \in M^r_{s,g,1} \cap M^r_{s,g,2}$ for every $s \in \mathcal{R}$ and $g \in \mathcal{E}$. If $s \neq r$, we have $M^r_{s,g,1} = M^r_{s,g,2} = \mathsf{Reg}(-2, 2, 1)$, and we have $(e_{s,g}, f_{s,g}) \in M^r_{s,g,1} \cap M^r_{s,g,2}$ since $e_{s,g}, f_{s,g} \in \{0, 1, 2\}$ by construction. Let us now consider the cases where $s = r$.

- If $g \neq e$, we have $e_{r,g} \in \{1, 2\}$ and thus $(e_{r,g}, f_{r,g}) \in M^r_{r,g,1} \cap M^r_{r,g,2}$, regardless of the value of $f_{r,g} \in \{0, 1, 2\}$.
- Now suppose $g = e$ with $e \neq f$. Then $e_{r,g} = e_{r,e} = 0$. Since $(e, r, f) \in \mathcal{G}$, we have $f_{r,g} = 1$. We thus have $(e_{r,g}, f_{r,g}) \in M^r_{r,g,1} \cap M^r_{r,g,2}$.
- Finally, suppose $g = e = f$. Since we assumed that $(e, r, f) = (e, r, e) \in \mathcal{G}$ we have that $M^r_{r,g,1} = M^r_{r,g,2} = \mathsf{Reg}(-2, 1, 1)$. We also have $e_{r,g} = f_{r,g} = e_{r,e} = 0$ and thus $(e_{r,g}, f_{r,g}) \in M^r_{r,g,1} \cap M^r_{r,g,2}$.

Now suppose $(e, r, f) \notin \mathcal{G}$. We need to show that there exists some $g \in \mathcal{E}$ such that $(e_{r,g}, f_{r,g}) \notin M^r_{r,g,1}$ or $(e_{r,g}, f_{r,g}) \notin M^r_{r,g,2}$. If $e = f$ we know that $M^r_{r,e,1} = \mathsf{Reg}(-2, 1, 1)$ and $M^r_{r,e,2} = \mathsf{Reg}(-4, -1, -1)$ since we assumed $(e, r, e) \notin \mathcal{G}$. Moreover, have $e_{r,e} = 0$ and thus $(e_{r,e}, e_{r,e}) \notin \mathsf{Reg}(-4, -1, -1)$. Now assume $e \neq f$. From the fact that $e_{r,e} = 0$ and $f_{r,e} = 2$, and the fact that $M^r_{r,e,1} = \mathsf{Reg}(-2, 1, 1)$, we find that $(e_{r,g}, f_{r,g}) \notin M^r_{r,e,1}$ holds for $g = e$.

A.2 Composition of MuRE regions

We consider a composition operation for two-dimensional regions, inspired by the definion of relational composition (Charpenay and Schockaert, 2024):

$$X_1 \diamond X_2 = \{(x, z) \mid \exists y \,.\, (x, y) \in X_1 \wedge (y, z) \in X_2\}$$

This composition operator will allow us to analyze under what conditions a rule is captured by a MuRE region embedding. Note that $\diamond$ is associative, i.e. $X_1 \diamond (X_2 \diamond X_3) = (X_1 \diamond X_2) \diamond X_3$. For MuRE regions, we can characterize the composition operator $\diamond$ as follows.

Proposition 4. *Let $u^-_1 \leq u^+_1$, $u^-_2 \leq u^+_2$ and $\lambda_1, \lambda_2 \in \mathbb{R}$. If $\lambda_1 > 0$, it holds that:*

$$\begin{aligned}&\mathsf{Reg}(u^-_1, u^+_1, \lambda_1) \diamond \mathsf{Reg}(u^-_2, u^+_2, \lambda_2)\\ &= \mathsf{Reg}(u^-_1 + \lambda_1 u^-_2, u^+_1 + \lambda_1 u^+_2, \lambda_1\lambda_2)\end{aligned}$$

Proof. Assume that (x, z) belongs to the region $\mathsf{Reg}(u^-_1, u^+_1, \lambda_1) \diamond \mathsf{Reg}(u^-_2, u^+_2, \lambda_2)$. Then there is some y such that

$$u^-_1 \leq \lambda_1 y - x \leq u^+_1 \tag{5}$$
$$u^-_2 \leq \lambda_2 z - y \leq u^+_2 \tag{6}$$

which implies

$$u^-_1 + \lambda_1 u^-_2 \leq \lambda_1\lambda_2 z - x \leq u^+_1 + \lambda_1 u^+_2 \tag{7}$$

Conversely, suppose (7) is satisfied. We need to show that we can always find a $y \in \mathbb{R}$ such that (5) and (6) are satisfied. Since we assumed $\lambda_1 > 0$, this is the case iff

$$y \in [\frac{u^-_1 + x}{\lambda_1}, \frac{u^+_1 + x}{\lambda_1}] \cap [\lambda_2 z - u^+_2, \lambda_2 z - u^-_2]$$

This intersection is non-empty iff the following two inequalities are satisfied:

$$\frac{u^-_1 + x}{\lambda_1} \leq \lambda_2 z - u^-_2$$
$$\lambda_2 z - u^+_2 \leq \frac{u^+_1 + x}{\lambda_1}$$

Given that $\lambda_1 > 0$, these inequalities are equivalent with (7). □

Proposition 5. *Let $u^-_1 \leq u^+_1$, $u^-_2 \leq u^+_2$ and $\lambda_1, \lambda_2 \in \mathbb{R}$. If $\lambda_1 < 0$, it holds that:*

$$\begin{aligned}&\mathsf{Reg}(u^-_1, u^+_1, \lambda_1) \diamond \mathsf{Reg}(u^-_2, u^+_2, \lambda_2)\\ &= \mathsf{Reg}(u^-_1 + \lambda_1 u^+_2, u^+_1 + \lambda_1 u^-_2, \lambda_1\lambda_2)\end{aligned}$$

Proof. Assume that (x, z) belongs to the region $\mathsf{Reg}(u^-_1, u^+_1, \lambda_1) \diamond \mathsf{Reg}(u^-_2, u^+_2, \lambda_2)$. Then there is some y such that

$$u^-_1 \leq \lambda_1 y - x \leq u^+_1 \tag{8}$$
$$u^-_2 \leq \lambda_2 z - y \leq u^+_2 \tag{9}$$

which implies

$$u^-_1 + \lambda_1 u^+_2 \leq \lambda_1\lambda_2 z - x \leq u^+_1 + \lambda_1 u^-_2 \tag{10}$$

Conversely, suppose (10) is satisfied. We need to show that we can always find a $y \in \mathbb{R}$ such that (8) and (9) are satisfied. Since we assumed $\lambda_1 < 0$, this is the case iff

$$y \in [\frac{u^+_1 + x}{\lambda_1}, \frac{u^-_1 + x}{\lambda_1}] \cap [\lambda_2 z - u^+_2, \lambda_2 z - u^-_2]$$

This intersection is non-empty iff the following two inequalities are satisfied:

$$\frac{u^+_1 + x}{\lambda_1} \leq \lambda_2 z - u^-_2$$
$$\lambda_2 z - u^+_2 \leq \frac{u^-_1 + x}{\lambda_1}$$

Given that $\lambda_1 < 0$, these inequalities are equivalent with

$$u^+_1 + x \geq \lambda_1\lambda_2 z - \lambda_1 u^-_2$$
$$\lambda_1\lambda_2 z - \lambda_1 u^+_2 \geq u^-_1 + x$$

which is clearly equivalent with (10). □

Proposition 6. *Let $u_1^- \leq u_1^+$, $u_2^- \leq u_2^+$ and $\lambda_2 \in \mathbb{R}$. It holds that:*

$$\begin{aligned}&\mathsf{Reg}(u_1^-, u_1^+, 0) \diamond \mathsf{Reg}(u_2^-, u_2^+, \lambda_2)\\ &= \mathsf{Reg}(u_1^-, u_1^+, 0)\end{aligned}$$

Proof. Assume that (x, z) belongs to the region $\mathsf{Reg}(u_1^-, u_1^+, 0) \diamond \mathsf{Reg}(u_2^-, u_2^+, \lambda_2)$. Then there is some y such that $(x, y) \in \mathsf{Reg}(u_1^-, u_1^+, 0)$, which imples $u_1^- \leq x \leq u_1^+$ and also $(x, z) \in \mathsf{Reg}(u_1^-, u_1^+, 0)$. Conversely, suppose $(x, z) \in \mathsf{Reg}(u_1^-, u_1^+, 0)$ holds, then we have $u_1^- \leq x \leq u_1^+$, which means that $(x, y) \in \mathsf{Reg}(u_1^-, u_1^+, 0)$ for any $y \in \mathbb{R}$. Let us choose $y = \lambda_2 z - u_2^-$. Then we also have $(y, z) \in \mathsf{Reg}(u_2^-, u_2^+, \lambda_2)$. It follows that $(x, z) \in \mathsf{Reg}(u_1^-, u_1^+, 0) \diamond \mathsf{Reg}(u_2^-, u_2^+, \lambda_2)$. □

The following result clarifies how the composition operator $\diamond$ allows us to check if a given rule is captured by a relation embedding.

Proposition 7. *Let $Z = Z_1 \times ... \times Z_n$. Suppose that $\tau_i(r_j) \cap Z_i^2 \neq \emptyset$ for all $i \in \{1, ..., n\}$ and $j \in \{1, ..., p\}$. The closed path rule $r_1(X_1, X_2) \wedge ... \wedge r_p(X_p, X_{p+1}) \rightarrow s(X_1, X_{p+1})$ is captured by the relation embedding τ iff for every $i \in \{1, ..., n\}$ we have:*

$$(\tau_i(r_1) \cap Z_i^2) \diamond ... \diamond (\tau_i(r_p) \cap Z_i^2) \subseteq \tau_i(s)$$

Proof. The assertion that the rule $r_1(X_1, X_2) \wedge ... \wedge r_p(X_p, X_{p+1}) \rightarrow s(X_1, X_{p+1})$ is captured by τ is equivalent, by definition, to the following assertion: for all $\mathbf{e_1}, ..., \mathbf{e_{p+1}}$ in Z such that $\tau \models r_1(\mathbf{e_1}, \mathbf{e_2}), ..., \tau \models r_p(\mathbf{e_p}, \mathbf{e_{p+1}})$, it holds that $\tau \models s(\mathbf{e_1}, \mathbf{e_{p+1}})$. In other words, for all $e_{ij} \in Z_i$, with $i \in \{1, ..., n\}$ and $j \in \{1, ..., p+1\}$, we have that the following holds: if for every $i \in \{1, ..., n\}$ we have $(e_{i1}, e_{i2}) \in \tau_i(r_1) \wedge ... \wedge (e_{ip}, e_{i(p+1)}) \in \tau_i(r_p)$, then for every $i \in \{1, ..., n\}$ we also have $(e_{i1}, e_{i(p+1)}) \in \tau_i(s)$. This is equivalent with the following assertion: if for every $i \in \{1, ..., n\}$ we have

$$(e_{i1}, e_{i(p+1)}) \in (\tau_i(r_1) \cap Z_i^2) \diamond ... \diamond (\tau_i(r_p) \cap Z_i^2)$$

then for every $i \in \{1, ..., n\}$ we also have $(e_{i1}, e_{i(p+1)}) \in \tau_i(s)$. Since we assumed $\tau_i(r_j) \cap Z_i^2 \neq \emptyset$ for all $i \in \{1, ..., n\}$ and $j \in \{1, ..., p\}$, we also have $\tau_i(r_1) \diamond ... \diamond \tau_i(r_p) \neq \emptyset$ for every $i \in \{1, ..., n\}$. The previous assertion is then equivalent with the condition that $(\tau_i(r_1) \cap Z_i^2) \diamond ... \diamond (\tau_i(r_p) \cap Z_i^2) \subseteq \tau_i(s)$ for every $i \in \{1, ..., n\}$. □

A.3 Negative Result on Composition

We show that MuRE regions cannot capture even simple sets of rules for $Z = \mathbb{R}^n$. In particular, we have the following result.

Proposition 8. *Let $\mathcal{K}$ consist of the following rules:*

$$\begin{aligned}r_1(X, Y) &\rightarrow s(X, Y)\\ r_2(X, Y) &\rightarrow s(X, Y)\\ r_1(X, Y) \wedge r_2(Y, Z) &\rightarrow s(X, Z)\end{aligned}$$

Let $Z = \mathbb{R}^n$. Any MuRE relation embedding which captures the rules in $\mathcal{K}$ also captures the rule $r_2(X, Y) \wedge r_1(Y, Z) \rightarrow s(X, Z)$.

Proof. A key observation to prove Proposition 8 is that we can only have $\mathsf{Reg}(u_1^-, u_1^+, \lambda_1) \subseteq \mathsf{Reg}(u_2^-, u_2^+, \lambda_2)$ if $\lambda_1 = \lambda_2$, which severely limits how regions can be captured.

Let $Z = \mathbb{R}^n$ and let $\mathcal{K}$ consist of the following rules:

$$\begin{aligned}r_1(X, Y) &\rightarrow s(X, Y)\\ r_2(X, Y) &\rightarrow s(X, Y)\\ r_1(X, Y) \wedge r_2(Y, Z) &\rightarrow s(X, Z)\end{aligned}$$

Suppose the relation embedding τ captures each of these rules. From Proposition 7 we then know that the following is true for every $i \in \{1, ..., n\}$:

$$\begin{aligned}\tau_i(r_1) &\subseteq \tau_i(s)\\ \tau_i(r_2) &\subseteq \tau_i(s)\\ \tau_i(r_1) \diamond \tau_i(r_2) &\subseteq \tau_i(s)\end{aligned}$$

Let $\tau_i(r_j) = \mathsf{Reg}(u_{r_j}^-, u_{r_j}^+, \lambda_{r_j})$ and $\tau_i(s) = \mathsf{Reg}(u_s^-, u_s^+, \lambda_s)$. From Propositions 4–6, we know that $\tau_i(r_1) \diamond \tau_i(r_2) = \mathsf{Reg}(u^-, u^+, \lambda_{r_1}\lambda_{r_2})$ for some $u^- \leq u^+$. From the first two rules we infer that:

$$\lambda_{r_1} = \lambda_{r_2} = \lambda_s$$

Let us write this value as λ. From the last rule, we infer:

$$\lambda^2 = \lambda$$

This is only possible if $\lambda = 1$ or $\lambda = 0$. If $\lambda = 0$, we have $\tau_i(r_2) \diamond \tau_i(r_1) = \mathsf{Reg}(u_{r_2}^-, u_{r_2}^+, 0) = \tau_i(r_2)$ and thus we find that

$$\tau_i(r_2) \diamond \tau_i(r_1) \subseteq \tau_i(s) \tag{11}$$

If $\lambda = 1$ we have $\tau_i(r_2) \diamond \tau_i(r_1) = \mathsf{Reg}(u_{r_1}^- + u_{r_2}^-, u_{r_1}^+ + u_{r_2}^+, 0) = \tau_i(r_1) \diamond \tau_i(r_2)$, hence we again find that (11) is satisfied. We thus have that (11) must be satisfied for every $i \in \{1, ..., n\}$, meaning that the rule $r_2(X, Y) \wedge r_1(Y, Z) \rightarrow s(X, Z)$ is captured by τ. □

A.4 Proof of Proposition 2

For the ease of presentation, in the folowing we will assume that $Z = [0,1]^n$. The following result shows that we can make this assumption w.l.o.g.

Lemma 1. *Let τ be a relation embedding, and let $Z' = [z_1^-, z_1^+] \times ... \times [z_n^-, z_n^+]$, with $z_i^- < z_i^+$ for all $i \in \{1, ..., n\}$. There exists a relation embedding τ' such that for any rule ρ we have that τ captures ρ with entity domain $[0,1]^n$ iff τ' captures ρ with entity domain Z'.*

Proof. Let us write $\tau_i(r) = \mathsf{Reg}(u_{r,i}^-, u_{r,i}^+, \lambda_{r,i})$, for each $r \in \mathcal{R}$. We define τ' as follows:

$$\begin{aligned}\tau_i'(r) = \mathsf{Reg}(&u_{r,i}^-(z_i^+ - z_i^-) + (\lambda_{r,i} - 1)z_i^-,\\ &u_{r,i}^+(z_i^+ - z_i^-) + (\lambda_{r,i} - 1)z_i^-,\\ &\lambda_{r,i})\end{aligned}$$

Noting that $z_i^+ > z_i^-$, we have that $u_{r,i}^- \leq \lambda_{r,i} y - x$ is equivalent to

$$\begin{aligned}&u_{r,i}^-(z_i^+ - z_i^-) + (\lambda_{r,i} - 1)z_i^-\\ &\leq \lambda_{r,i}(z_i^- + y(z_i^+ - z_i^-)) - (z_i^- + x(z_i^+ - z_i^-))\end{aligned}$$

and similar for the upper bond. We thus find that $(x,y) \in \tau_i(r)$ iff $(z_i^- + x(z_i^+ - z_i^-), z_i^- + y(z_i^+ - z_i^-)) \in \tau_i'(r)$, from which the result immediately follows. □

A.4.1 Bounded MuRE Regions

To check whether a given rule is captured by a relation embedding, we will again rely on Proposition 7. However, since $Z = [0,1]^n$ we cannot rely on Propositions 4–6 to characterize compositions of the form $(\tau(r_1) \cap Z) \diamond (\tau(r_2) \cap Z)$. However, as we will see, under some conditions, the characterization from Proposition 4 can still be used. Let us define:

$$\begin{aligned}\mathsf{BReg}(u,\lambda) = \{(x,y) \mid &0 \leq x \leq 1, 0 \leq y \leq 1,\\ &\lambda y - x \leq u\}\end{aligned} \tag{12}$$

The bounded region $\mathsf{BReg}(u,\lambda)$ corresponds to a region of the form $\mathsf{Reg}(u, u^+, \lambda) \cap [0,1]^2$, where the upper bound u^+ is always trivial. Note that we only consider trivial upper bounds because they play no role in the proof, and this simplifies the presentation. Similarly, we will only consider the case where $\lambda > 0$, as we will only rely on regions with positive scaling factors in the proof. We first characterize regions of the form $\mathsf{BReg}(u,\lambda)$ in terms of their vertices.

Lemma 2. *Let $\lambda > 0$ and $\lambda - 1 \leq u \leq 0$. It holds that $\mathsf{BReg}(u,\lambda)$ is equal to the following region*

$$CH\{(-u,0), (-u+\lambda, 1), (1,0), (1,1)\}$$

where we write CH for the convex hull operator.

Proof. First note that we clearly have

$$\begin{aligned}&\mathrm{CH}\{(-u,0), (-u+\lambda,1), (1,0), (1,1)\}\\ &\quad = \{(x,y) \mid x \leq 1, 0 \leq y \leq 1, \lambda y - x \leq u\}\end{aligned}$$

which already shows $\mathsf{BReg}(u,\lambda) \subseteq \mathrm{CH}\{(-u,0), (-u+\lambda,1), (1,0), (1,1)\}$.

Since $u \geq \lambda - 1 > -1$ and $u \leq 0$, it holds that $(-u, 0) \in [0,1]^2$. We also trivially have $\lambda \cdot 0 - (-u) \leq u$, and thus we have $(-u,0) \in \mathsf{BReg}(u,\lambda)$. Similarly, we have $-u + \lambda \leq (1-\lambda) + \lambda = 1$ and $-u + \lambda > -u \geq 0$, and thus $(-u+\lambda, 1) \in [0,1]^2$. Furthermore, we trivially have $\lambda - (-u+\lambda) \leq u$, and thus $(-u+\lambda, 1) \in \mathsf{BReg}(u,\lambda)$. We also trivially have $(1,0) \in \mathsf{BReg}(u,\lambda)$, nothing that $\lambda \cdot 0 - 1 = -1 \leq \lambda - 1 \leq u$, and $(1,1) \in \mathsf{BReg}(u,\lambda)$, noting that $\lambda - 1 \leq u$. We have thus also established that $\mathsf{BReg}(u,\lambda) \supseteq \mathrm{CH}\{(-u,0), (-u+\lambda,1), (1,0), (1,1)\}$. □

Proposition 9. *Let $\lambda_1, \lambda_2 > 0$, $\lambda_1 - 1 \leq u_1 \leq 0$ and $\lambda_2 - 1 \leq u_2 \leq 0$. It holds that:*

$$\begin{aligned}&\mathsf{BReg}(u_1,\lambda_1) \diamond \mathsf{BReg}(u_2,\lambda_2)\\ &\quad = \mathsf{BReg}(u_1 + \lambda_1 u_2, \lambda_1\lambda_2)\end{aligned}$$

Proof. Let $(x,z) \in \mathsf{BReg}(u_1,\lambda_1) \diamond \mathsf{BReg}(u_2,\lambda_2)$. Then there is some y such that

$$\lambda_1 y - x \leq u_1 \qquad \lambda_2 z - y \leq u_2 \tag{13}$$

which implies

$$\lambda_1\lambda_2 z - x \leq u_1 + \lambda_1 u_2 \tag{14}$$

Moreover, we also trivially have $x, z \in [0,1]$. It follows that $(x,z) \in \mathsf{BReg}(u_1 + \lambda_1 u_2, \lambda_1\lambda_2)$.

Conversely, suppose that $(x,z) \in \mathsf{BReg}(u_1 + \lambda_1 u_2, \lambda_1\lambda_2)$ holds. Then we have that (14) is satisfied. From Lemma 2, we furthermore know that $x \geq -u_1$. We need to show that we can always find a $y \in [0,1]$ such that the inequalities in (13) are satisfied. Since we assumed $\lambda_1 > 0$, this is the case iff

$$y \in \left[\lambda_2 z - u_2, \frac{u_1 + x}{\lambda_1}\right] \cap [0,1]$$

Since (14) is satisfied, we know that the first interval is always non-empty. To show that the intersection is non-empty, we need to show:

$$\frac{u_1 + x}{\lambda_1} \geq 0 \qquad \lambda_2 z - u_2 \leq 1$$

The first inequality follows from the fact that we have already established $x \geq -u_1$. The second inequality follows from the assumption that $u_2 \geq \lambda_1 - 1$ and the fact that $z \leq 1$. □

The closure of two regions $\mathsf{BReg}(u_1, \lambda_1)$ and $\mathsf{BReg}(u_2, \lambda_2)$ is defined as the smallest region of the form $\mathsf{BReg}(u_3, \lambda_3)$ that contains these two regions. We have the following result.

Proposition 10. *Let $\lambda_1 > 0$, $\lambda_2 > 0$, $\lambda_1 - 1 \leq u_1 \leq 0$ and $\lambda_2 - 1 \leq u_2 \leq 0$. It holds that:*

$$cl(\mathsf{BReg}(u_1, \lambda_1), \mathsf{BReg}(u_2, \lambda_2)) = \mathsf{BReg}(u_3, \lambda_3)$$

with

$$\begin{aligned} u_3 &= \max(u_1, u_2) \\ \lambda_3 &= \min(\lambda_1 - u_1, \lambda_2 - u_2) + \max(u_1, u_2) \end{aligned}$$

Proof. From Lemma 2 we know that:

$$\begin{aligned} cl(\mathsf{BReg}(u_1, \lambda_1) &= CH\{(-u_1, 0), (-u_1 + \lambda_1, 1), \\ &\qquad (1, 0), (1, 1)\} \\ cl(\mathsf{BReg}(u_2, \lambda_2) &= CH\{(-u_2, 0), (-u_2 + \lambda_2, 1), \\ &\qquad (1, 0), (1, 1)\} \end{aligned}$$

while $\mathsf{BReg}(\max(u_1, u_2), \min(\lambda_1 - u_1, \lambda_2 - u_2) + \max(u_1, u_2))$ is given by

$$\begin{aligned} CH\{ &\min(-u_1, -u_2), 0), \\ &(\min(\lambda_1 - u_1, \lambda_2 - u_2), 1), \\ &(1, 0), (1, 1)\} \end{aligned}$$

from which the result immediately follows. □

A.4.2 Preliminaries

We will work with regions of the following form:

$$A_{i,m,p} = \mathsf{BReg}(-m\varepsilon\lambda^{-i}, \lambda^p)$$

where $i, m, p \in \mathbb{N}$, and $\varepsilon > 0$ and $0 < \lambda < 1$ are sufficiently small constants, to ensure that the conditions of Proposition 9 are satisfied. In particular, in the following we will assume that $\lambda^p - 1 \leq -m\varepsilon\lambda^{-i}$ for all the values of m and p that are considered. Further constraints on how small ε and λ need to be will be made below.

Lemma 3. *It holds that*

$$A_{i,m_1,p_1} \diamond A_{i+p_1,m_2,p_2} = A_{i,m_1+m_2,p_1+p_2}$$

Proof. We find using Proposition 9:

$$\begin{aligned} &A_{i,m_1,p_1} \diamond A_{i+p_1,m_2,p_2} \\ &= \mathsf{BReg}(-m_1\varepsilon\lambda^{-i}, \lambda^{p_1}) \\ &\quad \diamond \mathsf{BReg}(-m_2\varepsilon\lambda^{-i-p_1}, \lambda^{p_2}) \\ &= \mathsf{BReg}(-m_1\varepsilon\lambda^{-i} - \lambda^{p_1}(m_2\varepsilon\lambda^{-i-p_1}), \lambda^{p_1+p_2}) \\ &= \mathsf{BReg}(-(m_1 + m_2)\varepsilon\lambda^{-i}, \lambda^{p_1+p_2}) \end{aligned}$$

□

As a special case, we find:

$$A_{1,1,1} \diamond A_{2,1,1} \diamond ... \diamond A_{k,1,1} = A_{1,k,k}$$

Lemma 4. *It holds that*

$$\begin{aligned} &A_{i,m_1,p_1} \diamond A_{j,m_2,p_2} \\ &= \mathsf{BReg}(-m_1\varepsilon\lambda^{-i} - m_2\varepsilon\lambda^{-j+p_1}, \lambda^{p_1+p_2}) \end{aligned}$$

Proof. We find using Proposition 9:

$$\begin{aligned} &A_{i,m_1,p_1} \diamond A_{j,m_2,p_2} \\ &= \mathsf{BReg}(-m_1\varepsilon\lambda^{-i}, \lambda^{p_1}) \diamond \mathsf{BReg}(-m_2\varepsilon\lambda^{-j}, \lambda^{p_2}) \\ &= \mathsf{BReg}(-m_1\varepsilon\lambda^{-i} - \lambda^{p_1}(m_2\varepsilon\lambda^{-j}), \lambda^{p_1+p_2}) \\ &= \mathsf{BReg}(-m_1\varepsilon\lambda^{-i} - m_2\varepsilon\lambda^{-j+p_1}, \lambda^{p_1+p_2}) \end{aligned}$$

□

Lemma 5. *Let $\pi : \{j, ..., j+l-1\} \to \{j, ..., j+l-1\}$ be a permutation, such that $\pi(j + u) \neq j + u$ for at least one $u \in \{0, ..., l-1\}$. It holds that $A_{\pi(j),1,1} \diamond ... \diamond A_{\pi(j+l-1),1,1} = \mathsf{BReg}(u^*, \lambda^*)$ with $u^* \leq -\varepsilon\lambda^{-j-1}$.*

Proof. Let u be the smallest index for which $\pi(j + u) \neq j + u$. If $u = 0$ then we have $A_{\pi(j),1,1} = A_{j',1,1}$ for some $j' > j$. We then have

$$u^* \leq -\varepsilon\lambda^{-j'} \leq -\varepsilon\lambda^{-j-1}$$

If $u > 0$ we find

$$\begin{aligned} A_{j,1,1} \diamond ... \diamond A_{j+u-1,1} \diamond A_{j+u,1,1} \\ = A_{j,u,u} \diamond A_{j+u',1,1} \\ = \mathsf{BReg}(u', \lambda') \end{aligned}$$

with $u' > u$. By Lemma 4 we have

$$\begin{aligned} u' &= -u\varepsilon\lambda^{-j} - \varepsilon\lambda^{-j-u'+u} \\ &\leq -\varepsilon\lambda^{-j-u'+u} \\ &\leq -\varepsilon\lambda^{-j-1} \end{aligned}$$

□

A.4.3 Capturing Rule Bases

For the ease of presentation, we write $r_1 \circ .... \circ r_p \subseteq s$ to denote the closed path rule $r_1(X_1, X_2) \wedge ... \wedge r_p(X_p, X_{p+1}) \rightarrow s(X_1, X_{p+1})$.

Let $\mathcal{K}$ be a set of regular closed path rules, and assume that any closed path rule entailed by $\mathcal{K}$ is either a trivial rule of the form $r \subseteq r$ or a regular rule. We construct a relation embedding capturing the rules in $\mathcal{K}$ as follows. Let us consider assignments α from $\mathcal{R}$ to $\{0, ..., |\mathcal{R}|\}$. Let $\mathcal{A}$ be the set of all such assignments. We consider embeddings with one coordinate for each of these assignments. Let us write α_i for the assignment associated with coordinate i. We furthermore write $M_{r,i}$ for the i^{th} coordinate of the MuRE region representing relation r, where $M_{r,i}$ is of the form (12). We define these regions as follows. First, for $r \in \mathcal{R}$, with $\alpha_i(r) = j$, we define:

$$M_{r,i} = \begin{cases} A_{j,1,1} & \text{if } j < |\mathcal{R}| \\ CH\{(0,0),(1,0)\} & \text{otherwise} \end{cases}$$

We assume that the constant $\varepsilon > 0$, which is used in the definition of $A_{1,1,\lambda}$, is small enough such that $|\mathcal{R}|\varepsilon\lambda^{-|\mathcal{R}|} + \lambda \leq 1$. Let us write $\mathsf{DC}(\mathcal{K})$ for the deductive closure of $\mathcal{K}$. More precisely, $\mathsf{DC}(\mathcal{K})$ is the set of all closed path rules which can be entailed from $\mathcal{K}$ and which are not trivial rules of the form $r \subseteq r$. We can then consider the following recursive definition, which we know to be well-defined thanks to the acyclic nature of regular rule bases (Charpenay and Schockaert, 2024):

$$M_{r,i} = cl\{M_{r,i} \cup \{M_{s_1,i} \diamond ... \diamond M_{s_k,i} \mid (s_1 \circ ... \circ s_k \subseteq r) \in \mathsf{DC}(\mathcal{K})\}\}$$

Let us write $\tau_{\mathcal{K}}$ for the relation embedding defined above. The following result immediately follows from the construction of $\tau_{\mathcal{K}}$.

Lemma 6. *It holds that $\tau_{\mathcal{K}}$ captures every rule in $\mathcal{K}$.*

We still need to show that $\tau_{\mathcal{K}}$ only captures the closed path rules which are entailed by $\mathcal{K}$. We will first show this for rules of the form $r_1 \circ \ldots \circ r_k \subseteq r$ where all of the relations $r_1, \ldots, r_k, r$ are distinct. Let us consider such a composition $r_1 \circ \ldots \circ r_k$, where $k \leq |\mathcal{R}| - 1$ and $r_i \neq r_j$ for $i \neq j$. We associate with such a composition the following assignment α_i.

$$\alpha_i(r) = \begin{cases} j & \text{if } r = r_j \\ |\mathcal{R}| & \text{otherwise} \end{cases}$$

For a region of the form $M = \mathsf{BReg}(u, \lambda)$ we define:

$$lo(M) = -u$$
$$uo(M) = -u + \lambda$$
$$sc(M) = \lambda$$

Note that these values respectively correspond to the lower offset (i.e. the lowest x-value for which $(x, 0) \in M$), the upper offset (i.e. the lowest x-value for which $(x, 1) \in M$) and the scale factor.

Lemma 7. *Assume $\lambda < 1$. It holds that one of the following is true:*

- *$\mathcal{K}$ entails a rule of the form $r_{\pi_1} \circ ... \circ r_{\pi_v} \subseteq s$, $sc(M_{s,i}) = \lambda^v$ and $lo(M_{s,i}) \geq \varepsilon\lambda^{-j}$ with $j \geq \max\{\pi_z - z + 1 \mid z \in \{1, ..., v\}\}$.*
- *$\mathcal{K}$ entails a rule of the form $r_{\pi_1} \circ ... \circ r_{\pi_v} \subseteq s$ and $sc(M_{s,i}) > \lambda^v$.*
- *$\mathcal{K}$ does not entail any rules of the form $r_{\pi_1} \circ ... \circ r_{\pi_v} \subseteq s$ and $M_{s,i} = CH\{(0,0),(1,0)\}$.*

Proof. We show this result by structural induction. First, assume that there are no rules of the form $t_1 \circ ... \circ t_u \subseteq s$ which are entailed by $\mathcal{K}$, apart from the trivial rule $s \subseteq s$. If $s \notin \{r_1, ..., r_k\}$ then we have that $M_{s,i} = \{(0,0),(1,0)\}$, and the result is trivially satisfied. If $s = r_j$ for some $j \in \{1, .., k\}$, we have $M_{s,i} = A_{j,1,1}$. We then have $lo(M_{s,i}) = \varepsilon\lambda^{-j}$ and $sc(M_{s,i}) = \lambda$, and thus we again have that the result is valid.

To show the inductive step, suppose that for each rule $t_1 \circ ... \circ t_p \subseteq s$ entailed by $\mathcal{K}$, the result has already been shown for $t_1, ..., t_p$. Let $t_1 \circ ... \circ t_p \subseteq s$ be a rule from $\mathcal{K}$. First suppose that for some $j \in \{1, ..., p\}$ it holds that $\mathcal{K}$ does not entail any rules of the form $r_{\pi_1} \circ ... \circ r_{\pi_v} \subseteq t_j$. Let us write $\mathcal{K}_3^s$ for this set of rules. By induction, we then have that $M_{t_j,i} = \{(0,0),(1,0)\}$. It then also follows that $M_{t_1,i} \diamond ... \diamond M_{t_p,i} = \{(0,0),(1,0)\}$. Now suppose that for each $j \in \{1, ..., p\}$, $\mathcal{K}$ entails a rule of the form $r_{\sigma_{1,j}} \circ ... \circ r_{\sigma_{v_j,j}} \subseteq t_j$. Then, because of the induction hypothesis, we know that there must exist such rules such that either:

- *$sc(M_{t_j,i}) = v_j$* and $\text{lo}(M_{t_j,i}) \geq \varepsilon\lambda^{-l}$ with $l \geq \max\{\sigma_{z,j} - z + 1 \mid z \in \{1, ..., v_j\}\}$; or
- *$sc(M_{t_j,i}) > v_j$.*

By induction we have $sc(M_{t_j}, i) \geq v_j$ for every j, from which it follows that $sc(M_{t_1,i} \diamond ... \diamond M_{t_p,i}) \geq$

$v_1+...+v_p$. If the second option is satisfied for any $j \in \{1,...,p\}$ then we clearly have $sc(M_{t_1,i} \diamond ... \diamond M_{t_p,i}) > v_1 + ... + v_p$. Let us write $\mathcal{K}_2^s$ for this set of rules. Now suppose the first option is satisfied for every $j \in \{1,...,p\}$. Let us write $\mathcal{K}_1^s$ for this set of rules. Then we have $sc(M_{t_1,i} \diamond ... \diamond M_{t_p,i}) = v_1 + ... + v_p$. Furthermore, we find:

$$\begin{aligned} &lo(M_{t_1,i} \diamond ... \diamond M_{t_p,i}) \\ &\quad = lo(M_{t_1,i}) + \sum_{z=2}^{p} \lambda^{v_1+...+v_{z-1}} lo(M_{t_z,i}) \end{aligned}$$

Using the induction hypothesis, we find and $lo(M_{t_1,i} \diamond ... \diamond M_{t_p,i}) \geq lo(M_{t_1,i}) \geq \varepsilon\lambda^{-\sigma_{z,1}+z-1}$ for $z \in \{1,...,v_1\}$. We also find for $q \in \{2,...,p\}$ and $z \in \{1,...,v_q\}$:

$$\begin{aligned} &lo(M_{t_1,i} \diamond ... \diamond M_{t_p,i}) \\ &\quad \geq \lambda^{v_1+...+v_{q-1}} lo(M_{t_q,i}) \\ &\quad \geq \lambda^{v_1+...+v_{q-1}} \varepsilon\lambda^{-\sigma_{z,q}+z-1} \\ &\quad = \varepsilon\lambda^{-\sigma_{z,q}+z+v_1+...+v_{q-1}-1} \end{aligned}$$

Because this result holds for every rule of the form $t_1 \circ ... \circ t_p \subseteq s$ in $\mathcal{K}$, it follows that $\text{sc}(M_{s,i}) = \lambda^v$ and $\text{lo}(M_{s,i}) \geq \varepsilon\lambda^{-j}$ with $j \geq \max\{\pi_z - z + 1 \,|\, z \in \{1,...,v\}\}$.

By construction, we have that $M_{s,i}$ is the closure of the regions $M_{t_1,i} \diamond ... \diamond M_{t_p,i}$, over all rules $t_1 \circ ... \circ t_p \subseteq s$ entailed by $\mathcal{K}$. If $\mathcal{K}_1^s = \mathcal{K}_2^s = \emptyset$, then it follows that $M_{s,i} = \{(0,0),(1,0)\}$, and the stated result is clearly satisfied. Otherwise, we have:

$$\begin{aligned} uo(M_{s,i}) = \min\{&uo(M_{t_1,i} \diamond ... \diamond M_{t_p,i}) \,| \\ &t_1 \circ ... \circ t_p \subseteq s \in \mathcal{K}_1^s \cup \mathcal{K}_2^s\} \\ lo(M_{s,i}) = \min\{&lo(M_{t_1,i} \diamond ... \diamond M_{t_p,i}) \,| \\ &t_1 \circ ... \circ t_p \subseteq s \in \mathcal{K}_1^s \cup \mathcal{K}_2^s\} \end{aligned}$$

Specifically, $uo(M_{s,i}) = uo(M_{t_1,i} \diamond ... \diamond M_{t_p,i})$ for some rule $\rho_u = t_1 \circ ... \circ t_p \subseteq s \in \mathcal{K}_1^s \cup \mathcal{K}_2^s$ and $lo(M_{s,i}) = \rho_l = lo(M_{t'_1,i} \diamond ... \diamond M_{t'_q,i})$ for some rule $t'_1 \circ ... \circ t'_q \subseteq s \in \mathcal{K}_1^s \cup \mathcal{K}_2^s$. We find:

- If $\rho_u = \rho_l$ and the rule comes from $\mathcal{K}_1^s$, then we have established that the first condition from the lemma must be satisfied.
- If ρ_u comes from $\mathcal{K}_2^s$, then we have that the second condition from the lemma must be satisfied.
- If $\rho_u \neq \rho_l$ (and it is not possible to choose ρ_u and ρ_l such that $\rho_u = \rho_l$) then we have that the second condition from the lemma must be satisfied. Indeed, we then have

$$\begin{aligned} sc(M_{s,i}) &= uo(M_{s,i}) - lo(M_{s,i}) \\ &= uo(M_{t_1,i} \diamond ... \diamond M_{t_p,i}) \\ &\qquad - lo(M_{t'_1,i} \diamond ... \diamond M_{t'_q,i}) \\ &> uo(M_{t_1,i} \diamond ... \diamond M_{t_p,i}) \\ &\qquad - lo(M_{t_1,i} \diamond ... \diamond M_{t_p,i}) \\ &= sc(M_{t_1,i} \diamond ... \diamond M_{t_p,i}) \end{aligned}$$

where we have established that $sc(M_{t_1,i} \diamond ... \diamond M_{t_p,i}) \geq \lambda^v$ for some rule of the form $r_{\pi_1} \circ ... \circ r_{\pi_v} \subseteq s$ entailed by $\mathcal{K}$.

In all cases, we thus find that the result is satisfied. □

Lemma 8. *Assume $\lambda < 1$. Let $r_1, \ldots, r_k, r \in \mathcal{R}$ be distinct relations and assume that $\mathcal{K} \not\models r_1 \circ \ldots \circ r_k \subseteq r$. Then $M_{r_1,i} \diamond \ldots \diamond M_{r_k,i} \not\subseteq M_{r,i}$ for α_i the assignment defined above.*

Proof. We clearly have:

$$M_{r_1,i} \diamond \ldots \diamond M_{r_k,i} \supseteq A_{1,1,1} \diamond \ldots \diamond A_{k,1,1}$$

where the right-hand side equals $A_{1,k,k}$. In particular, we have that $M_{r_1,i} \diamond \ldots \diamond M_{r_k,i}$ contains the point $(\varepsilon\lambda^{-1} + \lambda^k, 1)$. To conclude the proof we show that this point does not belong to $M_{r,i}$. Note that $(\varepsilon\lambda^{-1} + \lambda^k, 1) \in M_{r,i}$ is equivalent with the condition $uo(M_{r,i}) \leq \varepsilon\lambda^{-1} + \lambda^k$. By Lemma 7, we know that there are only two situations where this condition can be satisfied:

- There might be a rule of the form $r_{\pi_1} \circ ... \circ r_{\pi_v} \subseteq r$, such that $\text{sc}(M_{r,i}) = \lambda^v$ and $\text{lo}(M_{r,i}) \geq \varepsilon\lambda^{-j}$ with $j \geq \max\{\pi_z - z + 1 \,|\, z \in \{1,...,v\}\}$. To have $uo(M_{r,i}) \leq \varepsilon\lambda^{-1} + \lambda^k$, we need $v = k$ and $j = 1$. However, we can only have $j = 1$ if $(\pi_1, ..., \pi_k) = (1, ..., k)$. In other words, we can only have $uo(M_{r,i}) \leq \varepsilon\lambda^{-1} + \lambda^k$ if $\mathcal{K} \models r_1 \circ \ldots \circ r_k \subseteq r$, which was assumed not to be the case.
- There might be a rule of the form $r_{\pi_1} \circ ... \circ r_{\pi_v} \subseteq r$ such that $\text{sc}(M_{s,i}) > \lambda^v$. However, we can then only have $uo(M_{r,i}) \leq \varepsilon\lambda^{-1} + \lambda^k$ if $\lambda^v < \lambda^k$, meaning $v > k$. However, this is not possible given that $r_{\pi_1}, ..., r_{\pi_v}$ were assumed to be distinct relations from $\{r_1, ..., r_k\}$.

It follows that $uo(M_{r,i}) > \varepsilon\lambda^{-1} + \lambda^k$. □

Lemma 8 shows that $\tau_{\mathcal{K}}$ does not capture any unwanted rules of the form $r_1 \circ \ldots \circ r_k \subseteq r$ where $r_1, \ldots, r_k, r$ are distinct relations. We now show that the same is true for rules where $r_1, \ldots, r_k, r$ are not necessarily distinct. Note that when $r_1, \ldots, r_k, r$ are not all distinct, we always have $\mathcal{K} \not\models r_1 \circ \ldots \circ r_k \subseteq r$ (except for the trivial rule $r \subseteq r$), given our assumption about $\mathcal{K}$.

Lemma 9. *Let $\lambda < 1$. Let $s_1, \ldots, s_l \in \mathcal{R}$ be such that $s_p = s_q$ for some $p \neq q$. There is a coordinate i such that $M_{s_1,i} \diamond \ldots \diamond M_{s_l,i} \not\subseteq M_{r,i}$ for any relation $r \in \mathcal{R}$.*

Proof. Let $r_1, \ldots, r_k$ be the unique relations among $s_1, \ldots, s_l$, i.e. we have $\{r_1, \ldots, r_k\} = \{s_1, \ldots, s_l\}$ with $k < l$. Let us define the assignment α_i as follows:

$$\alpha_i(s) = \begin{cases} 0 & \text{if } s \in \{r_1, ..., r_k\} \\ |\mathcal{R}| & \text{otherwise} \end{cases}$$

Clearly we have

$$M_{s_1,i} \diamond \ldots \diamond M_{s_l,i} \supseteq A_{0,1,1} \diamond \ldots \diamond A_{0,1,1} = A_{0,l,l}$$

In particular $(\lambda^l, 1) \in M_{s_1,i} \diamond \ldots \diamond M_{s_l,i}$. In contrast, since $\mathcal{K}$ only entails closed path rules which are regular, it is easy to see that $\text{sc}(M_{t_1,i} \diamond \ldots \diamond M_{t_p,i}) \geq \lambda^k$ for any rule $t_1 \circ ... \circ t_p \subseteq r$ entailed by $\mathcal{K}$. Since $k < l$ and $\lambda < 1$ it follows that $(\lambda^l, 1) \notin M_{r,i}$. □

Lemma 10. *Let $s_1, \ldots, s_l, r \in \mathcal{R}$ be such that $r \in \{s_1, \ldots, s_l\}$ and $l \geq 2$. There is a coordinate i such that $M_{s_1,i} \diamond \ldots \diamond M_{s_l,i} \not\subseteq M_{r,i}$ for any relation $r \in \mathcal{R}$.*

Proof. Let $r_1, \ldots, r_k$ be the unique relations among $s_1, \ldots, s_l$, i.e. we have $\{r_1, \ldots, r_k\} = \{s_1, \ldots, s_l\}$ with $k \leq l$. Let us consider the same assignment α_i as in the proof of Lemma 9. We then again find that $(\lambda^l, 1) \in M_{s_1,i} \diamond \ldots \diamond M_{s_l,i}$. By construction, any rule of the form $r_{\pi_1} \circ ... \circ r_{\pi_v} \subseteq r$ entailed by $\mathcal{K}$ is such that $r_{\pi_1}, ..., r_{\pi_v}$ are distinct relations from $\{r_1, ..., r_k\} \setminus r$. It follows that $\text{sc}(M_{t_1,i} \diamond \ldots \diamond M_{t_p,i}) \geq \lambda^{k-1}$ for any rule $t_1 \circ ... \circ t_p \subseteq r$ entailed by $\mathcal{K}$, and in particular $(\lambda^l, 1) \notin M_{r,i}$. □

Proposition 2 now directly follows from Lemmas 6, 8, 9 and 10.

A.5 Proof of Proposition 3

Let $\mathcal{K}$ be a set of symmetry, inversion, hierarchy and intersection rules. We denote an intersection rule $r_1(X,Y) \wedge r_2(X,Y) \rightarrow r_3(X,Y)$ as $r_1 \cap r_2 \subseteq r_3$. We denote an inversion rule $r_1(X,Y) \rightarrow r_2(Y,X)$ as $r_1 \subseteq r_2^{inv}$ and similar for hierarchy and symmetry rules.

We construct a relation embedding $\tau_{\mathcal{K}}$ capturing these rules, without capturing any rules not entailed by $\mathcal{K}$. We consider assignments $\alpha : \mathcal{R} \rightarrow \{-2, -1, 0, 1, 2\}$ and write $\mathcal{A}$ for the set of all such assignments. We will construct embeddings with one coordinate for each assignment, writing α_i for the assignment associated with coordinate i. Let $M_{r,i}$ be the region representing relation r in coordinate i. We initialise these regions as follows:

$$M_{r,i}^{(0)} = \begin{cases} \mathsf{Reg}(0, \alpha_i(r), 1) & \text{if } \alpha_i(r) > 0 \\ \mathsf{Reg}(\alpha_i(r), 0, 1) & \text{if } \alpha_i(r) < 0 \\ \mathsf{Reg}(-2, 2, 1) & \text{if } \alpha_i(r) = 0 \end{cases}$$

We then apply the following update rules ($j \geq 1$):

$$\begin{aligned} M_{r,i}^{(j)} = cl\{&M_{r,i}^{(j-1)} \\ &\cup \{M_{s,i}^{(j)} \mid \mathcal{K} \models s \subseteq r\} \\ &\cup \{(M_{s,i}^{(j)})^{inv} \mid \mathcal{K} \models s \subseteq r^{-1}\} \\ &\cup \{M_{s,i}^{(j)} \cap M_{t,i}^{(j)} \mid \mathcal{K} \models s \cap t \subseteq r\}\} \end{aligned}$$

where for $M = \mathsf{Reg}(u^-, u^+, \lambda)$, with $\lambda \neq 0$, we write $M^{inv} = \mathsf{Reg}(-\frac{u^+}{\lambda}, -\frac{u^-}{\lambda}, \frac{1}{\lambda})$. Clearly, in each iteration, we have $M_{r,i}^{(j)} \supseteq M_{r,i}^{(j-1)}$. Furthermore, there are only finitely many values $M_{r,i}^{(j)}$ can take. This iterative process thus reaches a fixpoint after a finite number of steps. Let $M_{r,i}$ be the resulting regions, and let $\tau_{\mathcal{K}}$ be the associated region embedding. The proof that $\tau_{\mathcal{K}}$ captures the rules in $\mathcal{K}$, and only those rules, then follows in entirely the same way as the proof for octagon embeddings from (Charpenay and Schockaert, 2024).

A.6 Limitations of MuRE Regions

Symmetric relations can only be modelled using regions $\mathsf{Reg}(u^-, u^+, \lambda)$ that satisfy either of the following conditions:

- $\lambda = 1$ and $u^- = -u^+$;
- $\lambda = -1$

This restriction entails various limitations. To illustrate this, first note that using Propositions 4 and 5,

we straightforwardly find:

$$\begin{aligned}&\mathsf{Reg}(-u_1, u_1, 1) \diamond \mathsf{Reg}(-u_2, u_2, 1)\\ &= \mathsf{Reg}(-u_1 - u_2, u_1 + u_2, 1)\\ &= \mathsf{Reg}(-u_2, u_2, 1) \diamond \mathsf{Reg}(-u_1, u_1, 1)\end{aligned}$$

$$\begin{aligned}&\mathsf{Reg}(-u_1, u_1, 1) \diamond \mathsf{Reg}(u_2^-, u_2^+, -1)\\ &= \mathsf{Reg}(-u_1 + u_2^-, u_1 + u_2^+, -1)\\ &= \mathsf{Reg}(u_2^-, u_2^+, -1) \diamond \mathsf{Reg}(-u_1, u_1, 1)\end{aligned}$$

Now suppose the relations r and s are symmetric, while $r \circ s \neq s \circ r$. It follows that there needs to be at least one coordinate in which the scaling factor is -1, for both r and s. However, for transitive relations, we clearly cannot have -1 as a scaling factor. It follows that when r and s are symmetric and r is transitive, whenever a rule $r \circ s \subseteq t$ is captured, then we also have that $s \circ r \subseteq t$ is captured, and vice versa.

B Additional Experimental Details and Analysis

B.1 Model Combinations

The various KG embedding models we have reviewed can be generated from five independent design choices:

- The dissimilarity measure which is used, where we consider two options: Euclidean distance and squared norm.
- Whether and which kinds of learnable bias terms are used. Here we consider four options: norms, entity biases, relation biases and a fixed margin.
- Whether scaling and/or translation are used, which leads to three options: only scaling, only translation, or both.
- Whether coordinate-wise and/or cross-coordinate comparisons are used, where there are three options: only coordinate-wise, only cross-coordinate or both.
- Whether a region-based view is adopted, with a learnable width parameter, which leads to two options: with or without a width parameter.

From these design choices, we obtain $2 \times 4 \times 3 \times 3 \times 2 = 144$ model combinations.

	d	$\lvert B\rvert$	δ	$\lvert N\rvert$	λ
Countries	40	256	10^{-2}	10	3
Kinships	40	256	10^{-2}	10	3
UMLS	40	256	10^{-2}	10	3
CPR	40	256	10^{-1}	10	3
CPR-rt	40	256	10^{-1}	10	3
CPR-rtu	40	256	10^{-1}	10	3
WN18RR	100	256	10^{-3}	50	6
FB15k-237	200	256	10^{-3}	50	9

Table 5: Hyperparameters used per dataset for training, where d is the embedding dimension, $|B|$ is the size of a training batch, δ is the learning rate, $|N|$ is the number of negative examples per triple and λ is the margin (for configurations without a learnable bias).

B.2 Hyperparameters

All results given in the paper were obtained by setting the hyperparameters listed in Table 5. These hyperparameters were identical for all models, other model-specific configuration points were left unchanged. The reported results for every model are those obtained with the version of the model that has the highest hits@10 on a validation KG (evaluation performed every 10 epochs). The maximum number of epochs is set to 500, training stops as soon as hits@10 on the validation KG decreases (no patience).

B.3 Small-scale Experiments

Tables 6–8 provide the detailed results we obtained for Countries, Kinships and UMLS. Note how the Countries dataset, designed to be noise-free, shows the highest variability. In the following, we comment the influence of each of the five design choices identified in the paper.

Dissimilarity Function As made clear in (1), models that are based on a dot product (i.e. comparison function h_2) can equivalently be characterized in terms of the Euclidean norm. We empirically compare these two formulations in the first block of results in Table 1. Despite their mathematical equivalence, they behave differently during optimisation, which might translate into a different empirical performance. We can indeed see some small differences between the performance of these two formulations, although they may not be significant, as the differences are of the same order of magnitude as the standard deviation across multiple runs. What is clearer, however, is the importance of the bias. The model that uses the squared norm and a

	Countries			
	h@1	h@3	h@10	MRR
TransE	0.262 ± 0.177	0.441 ± 0.045	0.520 ± 0.014	0.367 ± 0.109
RotatE	0.400 ± 0.027	0.979 ± 0.020	0.995 ± 0.009	0.670 ± 0.019
BoxE	0.804 ± 0.152	0.945 ± 0.060	1.0 ± 0.0	0.880 ± 0.095
MuRE	0.675 ± 0.078	0.979 ± 0.020	1.0 ± 0.0	0.820 ± 0.052
func. ExpressivE	0.512 ± 0.050	0.945 ± 0.056	1.0 ± 0.0	0.723 ± 0.014
ExpressivE	0.466 ± 0.230	0.712 ± 0.309	0.850 ± 0.266	0.609 ± 0.251
ComplEx	0.0 ± 0.0	0.008 ± 0.011	0.024 ± 0.017	0.016 ± 0.003
SimplE	0.104 ± 0.070	0.275 ± 0.205	0.504 ± 0.315	0.234 ± 0.150
$g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) \cdot \mathbf{f}$	0.404 ± 0.034	0.762 ± 0.011	0.945 ± 0.056	0.602 ± 0.020
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\|^2 + \|g_{\mathbf{s};\mathbf{u}}(\mathbf{e})\|^2 + \|\mathbf{f}\|^2$	0.341 ± 0.119	0.641 ± 0.114	0.837 ± 0.049	0.517 ± 0.098
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\|^2 + \lambda$	0.504 ± 0.063	0.870 ± 0.126	0.945 ± 0.058	0.688 ± 0.020
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + \lambda$	0.608 ± 0.074	0.962 ± 0.061	1.0 ± 0.0	0.790 ± 0.044
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_h + b_t$	0.720 ± 0.114	0.891 ± 0.101	0.937 ± 0.064	0.809 ± 0.097
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.695 ± 0.100	0.954 ± 0.069	0.983 ± 0.027	0.825 ± 0.070
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.729 ± 0.070	0.979 ± 0.029	1.0 ± 0.0	0.853 ± 0.044
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.683 ± 0.071	0.995 ± 0.009	0.995 ± 0.009	0.828 ± 0.030
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.662 ± 0.053	0.920 ± 0.092	0.991 ± 0.018	0.796 ± 0.036
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.674 ± 0.149	0.987 ± 0.027	1.0 ± 0.0	0.825 ± 0.085
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.650 ± 0.064	0.983 ± 0.037	0.991 ± 0.018	0.809 ± 0.045
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	0.633 ± 0.086	0.966 ± 0.027	0.991 ± 0.011	0.796 ± 0.042
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	0.620 ± 0.099	0.937 ± 0.046	1.0 ± 0.0	0.778 ± 0.062
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e}), \mathbf{f}) + b_r$	0.495 ± 0.082	0.770 ± 0.075	0.916 ± 0.048	0.644 ± 0.067
$-d_{\mathbf{w}}(g_{\mathbf{1};\mathbf{u}}(\mathbf{e}), \mathbf{f}) + b_r$	0.516 ± 0.082	0.945 ± 0.043	1.0 ± 0.0	0.717 ± 0.049
$-d_{\mathbf{w}}(g_{\mathbf{1};\mathbf{u}}(\mathbf{e};\mathbf{e}'), \mathbf{f}';\mathbf{f}) + b_r$	0.487 ± 0.089	0.883 ± 0.045	0.975 ± 0.022	0.686 ± 0.053
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}'), \mathbf{f}';\mathbf{f}) + b_r$	0.395 ± 0.084	0.616 ± 0.078	0.800 ± 0.066	0.537 ± 0.067
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}'), \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}) + b_r$	0.458 ± 0.159	0.691 ± 0.120	0.891 ± 0.075	0.604 ± 0.123

Table 6: Performance of link prediction on the Countries dataset (mean and standard deviation over 5 runs).

fixed bias ($\lambda = 3$) outperforms both variants, on all runs and all datasets. MRR increases by 2-8%, for example. Most linear models define d as the Euclidian distance. The configuration where λ is fixed and d is the Euclidian distance has an small edge over other configurations. We take it as a reference in the remainder of the section.

Trainable Biases Balazevic et al. (2019a) already observed that using trainable biases can lead to better results. However, they only evaluated entity-specific biases. In the second block of results in Table 1, we compare MuRE with models where d is the Euclidian distance and the bias term is (i) a learned constant λ (as previously introduced); (ii) a relation-specific term b_r; or (iii) an entity-specific term $b_e + b_f$. The use of relation-specific biases (b_r) tends to give the best results across the three datasets. Interestingly, on Countries, we found that the entity-specific bias terms b_e and b_f are strongly correlated with the degree of entities in the KG, being higher for entities that appear in more triples. Biases thus help capture a statistical prior for each entity. Such a prior does not help to predict links if the KG is balanced (as in Kinships, where the correlation between entity biases and degrees is close to 0) or if entities are highly connected (as in UMLS). More details can be found in Appendix B.5.

Scaling and Translation Most models transform the head entity either through scaling or translation but not both, with MuRE and ExpressivE being the most notable exceptions. An analysis across the three datasets shows, however, that combining scaling and translation is important (third block of results in Table 1). Translation alone underperforms on Kinships and UMLS. Moreover, combining translation and scaling improves MRR by 2-3% w.r.t. using scaling alone. On Countries, using translation along works well. However, recall that this is a simple and artificially constructed KG. In fact, it is sufficient to learn a single regular closed-path rule (which translational models can capture) to achieve strong link prediction results for this dataset. It is worth noting that some models in the literature also allow transformation through rotation[4]. RESCAL, for instance, embeds relations as arbitrary linear transformations, combining rotation with scaling (Nickel et al., 2011b). Later models extended it to affine transformations, also

[4]We refer here to rotation in $\mathbb{R}^n$. Note that RotatE, despite its name, should rather be considered as a scaling model: rotation is not in $\mathbb{R}^n$ but in the two-dimensional Euler plane.

	Kinships			
	h@1	h@3	h@10	MRR
TransE	0.026 ± 0.002	0.069 ± 0.004	0.190 ± 0.003	0.089 ± 0.003
RotatE	0.553 ± 0.015	0.810 ± 0.013	0.955 ± 0.001	0.697 ± 0.012
BoxE	0.517 ± 0.014	0.782 ± 0.015	0.957 ± 0.003	0.669 ± 0.011
MuRE	0.478 ± 0.021	0.736 ± 0.016	0.945 ± 0.004	0.633 ± 0.016
func. ExpressivE	0.521 ± 0.016	0.767 ± 0.008	0.938 ± 0.003	0.665 ± 0.010
ExpressivE	0.584 ± 0.007	0.821 ± 0.015	0.965 ± 0.002	0.718 ± 0.007
ComplEx	0.633 ± 0.017	0.857 ± 0.014	0.971 ± 0.001	0.757 ± 0.013
SimplE	0.343 ± 0.024	0.582 ± 0.014	0.875 ± 0.007	0.506 ± 0.018
$g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) \cdot \mathbf{f}$	0.418 ± 0.022	0.696 ± 0.021	0.940 ± 0.006	0.588 ± 0.018
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\|^2 + \|g_{\mathbf{s};\mathbf{u}}(\mathbf{e})\|^2 + \|\mathbf{f}\|^2$	0.434 ± 0.019	0.702 ± 0.022	0.939 ± 0.003	0.600 ± 0.017
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\|^2 + \lambda$	0.476 ± 0.007	0.741 ± 0.007	0.943 ± 0.003	0.632 ± 0.005
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + \lambda$	0.492 ± 0.021	0.745 ± 0.009	0.937 ± 0.003	0.643 ± 0.013
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_h + b_t$	0.490 ± 0.017	0.748 ± 0.016	0.944 ± 0.003	0.643 ± 0.014
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.467 ± 0.005	0.722 ± 0.017	0.928 ± 0.003	0.621 ± 0.007
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.071 ± 0.035	0.146 ± 0.038	0.304 ± 0.017	0.151 ± 0.032
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.496 ± 0.024	0.745 ± 0.018	0.939 ± 0.011	0.645 ± 0.018
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.631 ± 0.018	0.855 ± 0.012	0.968 ± 0.006	0.755 ± 0.013
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.074 ± 0.009	0.153 ± 0.015	0.321 ± 0.025	0.159 ± 0.009
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.633 ± 0.025	0.861 ± 0.013	0.971 ± 0.002	0.757 ± 0.018
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	0.609 ± 0.023	0.847 ± 0.012	0.973 ± 0.004	0.741 ± 0.018
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	0.658 ± 0.027	0.880 ± 0.021	0.978 ± 0.003	0.778 ± 0.020
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e}), \mathbf{f}) + b_r$	0.520 ± 0.014	0.784 ± 0.015	0.959 ± 0.004	0.673 ± 0.012
$-d_{\mathbf{w}}(g_{\mathbf{1};\mathbf{u}}(\mathbf{e}), \mathbf{f}) + b_r$	0.291 ± 0.052	0.598 ± 0.027	0.909 ± 0.007	0.487 ± 0.036
$-d_{\mathbf{w}}(g_{\mathbf{1};\mathbf{u}}(\mathbf{e};\mathbf{e}'), \mathbf{f}';\mathbf{f}) + b_r$	0.531 ± 0.014	0.800 ± 0.012	0.965 ± 0.003	0.683 ± 0.011
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}'), \mathbf{f}';\mathbf{f}) + b_r$	0.539 ± 0.029	0.798 ± 0.029	0.963 ± 0.007	0.686 ± 0.024
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}'), \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}) + b_r$	0.596 ± 0.014	0.850 ± 0.007	0.974 ± 0.002	0.735 ± 0.010

Table 7: Performance of link prediction on the Kinships dataset (mean and standard deviation over 5 runs).

featuring translation (Jiang et al., 2024; Ge et al., 2022). Yet, despite a significantly higher number of trainable parameters, the performance of these models does not exceed that of MuRE or RotatE, suggesting that scaling and translation are sufficient for faithful relation embeddings.

Cross-Coordinate Comparison We analyze models with cross-coordinate comparisons in the fourth block of results in Table 1. Having both coordinate-wise and cross-coordinate comparisons, as in ComplEx, slightly increases results, except for Countries. As already mentioned, this dataset requires learning a particular closed-path rule, which cross-coordinate comparison models struggle with. Conversely, for Kinships, using only cross-coordinate comparisons significantly improves results. Overall, our results highlight the usefulness of cross-coordinate comparisons. It is interesting to note, however, that ComplEx underperforms its linear variant. The gap is particularly large on Countries, which is likely due to the regularization scheme of ComplEx (which is also used by SimplE) being ineffective on small datasets.

Region-based Formulation BoxE and ExpressivE, as region based models, have a width parameter $\mathbf{w}$. The last block of results in Table 1 shows that such width-dependent scoring decreases the performance of the cross-coordinate model. It also decreases performance on Countries and UMLS for the baseline model. However, BoxE performs better than a baseline translational model on Kinships. ExpressivE performs best on Kinships and UMLS. Further experiments show that functional ExpressivE ($\mathbf{w} = \mathbf{0}$) outperforms MuRE on all datasets, including Countries.

B.4 License Details

All datasets have been used fully in line with their intended purpose. They can be found on Github[5] and via PyKEEN[6]. Note that PyKEEN only exposes the S1 version of Countries, which is the one we used. WN18RR is a subset of WordNet[7], available under a dedicated license (the WordNet license). FB15k-237 is a subset of Freebase, a large KG available under a CC-BY 2.5 license[8]. Countries, in its original form[9], is published under

[5] https://github.com/ZhenfengLei/KGDatasets

[6] https://pykeen.readthedocs.io/en/stable/reference/datasets.html

[7] https://wordnet.princeton.edu

[8] https://developers.google.com/freebase

[9] https://github.com/mledoze/countries

	UMLS			
	h@1	h@3	h@10	MRR
TransE	0.475 ± 0.016	0.650 ± 0.020	0.798 ± 0.004	0.588 ± 0.009
RotatE	0.668 ± 0.009	0.931 ± 0.004	0.982 ± 0.003	0.804 ± 0.005
BoxE	0.766 ± 0.019	0.958 ± 0.006	0.984 ± 0.003	0.865 ± 0.011
MuRE	0.775 ± 0.013	0.977 ± 0.006	0.996 ± 0.001	0.877 ± 0.008
func. ExpressivE	0.794 ± 0.018	0.982 ± 0.002	0.996 ± 0.000	0.888 ± 0.009
ExpressivE	0.816 ± 0.012	0.970 ± 0.004	0.993 ± 0.002	0.895 ± 0.006
ComplEx	0.774 ± 0.016	0.949 ± 0.011	0.985 ± 0.005	0.865 ± 0.011
SimplE	0.555 ± 0.028	0.722 ± 0.007	0.883 ± 0.004	0.664 ± 0.016
$g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) \cdot \mathbf{f}$	0.674 ± 0.012	0.935 ± 0.007	0.983 ± 0.003	0.808 ± 0.010
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\|^2 + \|g_{\mathbf{s};\mathbf{u}}(\mathbf{e})\|^2 + \|\mathbf{f}\|^2$	0.703 ± 0.035	0.932 ± 0.015	0.986 ± 0.001	0.823 ± 0.022
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\|^2 + \lambda$	0.760 ± 0.006	0.969 ± 0.007	0.994 ± 0.000	0.865 ± 0.003
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + \lambda$	0.767 ± 0.018	0.962 ± 0.013	0.989 ± 0.005	0.867 ± 0.014
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_h + b_t$	0.764 ± 0.016	0.967 ± 0.006	0.994 ± 0.001	0.867 ± 0.010
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.742 ± 0.020	0.940 ± 0.005	0.984 ± 0.002	0.846 ± 0.012
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.479 ± 0.012	0.812 ± 0.017	0.943 ± 0.007	0.660 ± 0.007
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.772 ± 0.017	0.971 ± 0.001	0.993 ± 0.000	0.873 ± 0.008
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.743 ± 0.036	0.934 ± 0.019	0.979 ± 0.006	0.844 ± 0.025
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.684 ± 0.016	0.921 ± 0.005	0.985 ± 0.002	0.811 ± 0.009
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.763 ± 0.016	0.961 ± 0.009	0.988 ± 0.004	0.865 ± 0.006
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	0.736 ± 0.031	0.936 ± 0.017	0.985 ± 0.006	0.841 ± 0.021
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	0.772 ± 0.004	0.967 ± 0.002	0.993 ± 0.001	0.872 ± 0.002
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e}), \mathbf{f}) + b_r$	0.767 ± 0.013	0.957 ± 0.003	0.985 ± 0.002	0.863 ± 0.007
$-d_{\mathbf{w}}(g_{\mathbf{1};\mathbf{u}}(\mathbf{e}), \mathbf{f}) + b_r$	0.548 ± 0.027	0.956 ± 0.005	0.990 ± 0.001	0.753 ± 0.013
$-d_{\mathbf{w}}(g_{\mathbf{1};\mathbf{u}}(\mathbf{e};\mathbf{e}'), \mathbf{f}';\mathbf{f}) + b_r$	0.754 ± 0.028	0.957 ± 0.010	0.985 ± 0.003	0.859 ± 0.018
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}'), \mathbf{f}';\mathbf{f}) + b_r$	0.739 ± 0.016	0.950 ± 0.003	0.983 ± 0.005	0.848 ± 0.006
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}'), \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}) + b_r$	0.768 ± 0.024	0.965 ± 0.007	0.992 ± 0.003	0.869 ± 0.012

Table 8: Performance of link prediction on the UMLS dataset (mean and standard deviation over 5 runs).

an ODC Open Database license (ODbl). These licenses permit use, copy and modification. UMLS is licensed by the US National Medical Library[10]. Some restrictions apply on its use in general, but not in academia. No license information is known for the Kinships dataset. It is original work by the anthropologist Woodrow W. Denham.

B.5 Trainable Biases

We analyze to what extent entity biases encode the importance of entities in the KG, where we interpret importance in terms of their in-degree. Figures 1, 2 and 3 show how the in-degree of an entity relates to the size of its corresponding bias. These plots were produced with the biases learnt with the following scoring function:

$$\|\mathbf{e} \odot \mathbf{s} + \mathbf{u} - \mathbf{f}\| + b_e + b_f$$

where $\mathbf{e}, \mathbf{s}, \mathbf{u}, \mathbf{f} \in \mathbb{R}^{40}$. In Countries (Figure 1), the correlation between in-degree and learnt bias is strong. In UMLS (Figure 3), a correlation also exists, but it is not as strong. The range of in-degrees is also much broader in this case. Kinships is the exception (Figure 2). All entities have roughly the same in-degree (∼80), yet we still find bias terms

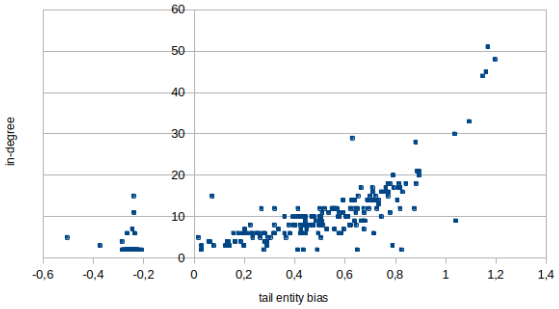

Figure 1: In-degree of entities vs. learnt entity biases for Countries.

ranging from around 1 to around 2, which appear to be the result of overfitting.

B.6 Synthetic Experiments

Characteristics of our synthetic closed-path rule datasets can be found in Table 9. Tables 10–12 provide the detailed results we obtained for these datasets.

B.7 Benchmark Experiments

Experimental Settings Results found in the literature on WN18RR and FB15k-237 are not directly comparable to each other. For instance, RotatE and ComplEx, whose entity embeddings are defined in

[10] https://www.nlm.nih.gov/databases/umls.html

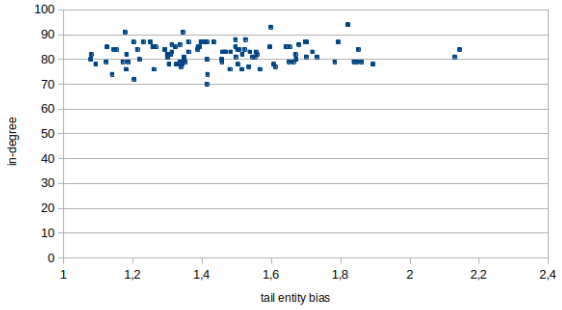

Figure 2: In-degree of entities vs. learnt entity biases for Kinships.

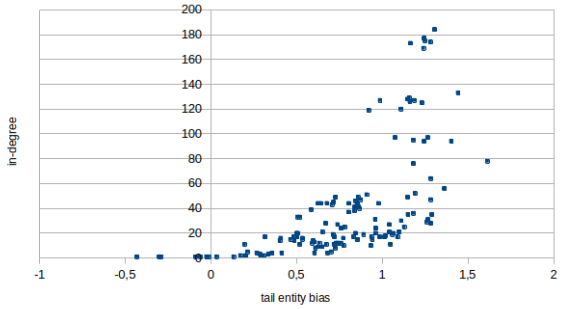

Figure 3: In-degree of entities vs. learnt entity biases for UMLS.

$\mathbb{C}^n$, have been compared to TransE and DistMult models with embeddings in $\mathbb{R}^n$ while they should in fact be compared with models in $\mathbb{R}^{2n}$. Similarly, BoxE should be compared with models that have embeddings in $\mathbb{R}^{2n}$ if BoxE embeddings are of the form $[\mathbf{e}; \mathbf{e}'] \in \mathbb{R}^{2n}$. As shown in our main experiments (Tables 1, 2 and 3), cross-coordinate comparisons and other interdependencies across coordinates increase the expressivity of base models even with fixed dimensions.

Table 13 provides further experimental results on WN18RR and FB15k-237, comparing our baseline models with state-of-the-art models, regardless of the various mismatches we could find regarding dimensionality and negative sampling. The results for existing models are those reported by Pavlovic and Sallinger (2023) for ExpressivE, Abboud et al. (2020) for BoxE, Sun et al. (2019) for RotatE, Lacroix et al. (2018) for ComplEx[11] and Zhang et al. (2019a) for QuatE. All results were obtained with $n = 500$ for WN18RR and $n = 1000$ for FB15k-237, except for QuatE. Since no results were available for QuatE with these dimensions, its

[11] full results for ComplEx are not given in the paper but they are available at https://github.com/facebookresearch/kbc.

	$\lvert\mathcal{E}\rvert$	$\mathcal{R}\rvert$	$\lvert\mathcal{G}_{\text{train}}\rvert$	$\lvert\mathcal{G}_{\text{eval}}\rvert$
CPR	750	3	1110	220
CPR-rt	750	3	1955	906
CPR-rtu	750	4	1959	492

Table 9: Characteristics of synthetic datasets.

results are given for $n = 100$ for both WN18RR and FB15k-237. MuRE variants were all trained with inverse relations.

Computation Time Table 14 compares the training time of different variants of the models. The baseline models $-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}) + b_r$, and to a lesser extent $-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$, stand out as the most efficient. ExpressivE, on the contrary, is the least efficient. ComplEx is the second least efficient. ComplEx and QuatE have the same regularization scheme but the latter converges much faster than the former. Morever, state-of-the-art results of ComplEx come with an even higher computational cost because they are obtained in a 1-vs-all setting. Other models with cross-coordinate comparisons in Table 14 roughly double the computation time of the baseline models. RotatE is more efficient than these models, while still less efficient than the baseline model without inverse relations.

	CPR			
	h@1	h@3	h@10	MRR
TransE	0.009 ± 0.004	0.025 ± 0.009	0.075 ± 0.014	0.036 ± 0.006
RotatE	0.743 ± 0.032	0.917 ± 0.024	0.983 ± 0.012	0.834 ± 0.020
BoxE	0.014 ± 0.007	0.042 ± 0.008	0.107 ± 0.035	0.046 ± 0.011
MuRE	0.346 ± 0.051	0.534 ± 0.052	0.751 ± 0.068	0.477 ± 0.049
ComplEx	0.021 ± 0.026	0.039 ± 0.049	0.089 ± 0.092	0.048 ± 0.048
SimplE	0.021 ± 0.012	0.064 ± 0.032	0.165 ± 0.063	0.071 ± 0.026
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.282 ± 0.044	0.480 ± 0.078	0.657 ± 0.082	0.411 ± 0.057
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.015 ± 0.005	0.121 ± 0.022	0.656 ± 0.030	0.164 ± 0.008
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.252 ± 0.042	0.432 ± 0.045	0.613 ± 0.052	0.375 ± 0.042
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.111 ± 0.014	0.173 ± 0.025	0.271 ± 0.039	0.163 ± 0.017
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.003 ± 0.005	0.037 ± 0.009	0.189 ± 0.024	0.058 ± 0.007
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.115 ± 0.015	0.175 ± 0.018	0.247 ± 0.021	0.163 ± 0.012
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	0.316 ± 0.070	0.501 ± 0.099	0.717 ± 0.120	0.446 ± 0.085
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	0.309 ± 0.105	0.450 ± 0.133	0.648 ± 0.157	0.417 ± 0.118

Table 10: Performance of link prediction on the closed-path rule (CPR) synthetic dataset (mean and standard deviation over 5 runs).

	CPR-rt			
	h@1	h@3	h@10	MRR
TransE	0.017 ± 0.005	0.038 ± 0.006	0.087 ± 0.010	0.047 ± 0.005
RotatE	0.671 ± 0.016	0.859 ± 0.011	0.964 ± 0.004	0.777 ± 0.011
BoxE	0.011 ± 0.001	0.046 ± 0.008	0.149 ± 0.014	0.057 ± 0.004
MuRE	0.276 ± 0.015	0.524 ± 0.028	0.765 ± 0.026	0.438 ± 0.020
ComplEx	0.117 ± 0.014	0.224 ± 0.022	0.410 ± 0.035	0.211 ± 0.018
SimplE	0.088 ± 0.017	0.209 ± 0.028	0.408 ± 0.035	0.190 ± 0.022
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.332 ± 0.033	0.655 ± 0.049	0.899 ± 0.045	0.524 ± 0.037
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.002 ± 0.002	0.053 ± 0.006	0.520 ± 0.010	0.121 ± 0.003
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.312 ± 0.022	0.624 ± 0.037	0.871 ± 0.024	0.501 ± 0.025
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.154 ± 0.019	0.274 ± 0.017	0.440 ± 0.031	0.248 ± 0.020
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.005 ± 0.001	0.042 ± 0.003	0.294 ± 0.014	0.083 ± 0.001
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.150 ± 0.026	0.249 ± 0.042	0.394 ± 0.048	0.231 ± 0.035
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	0.358 ± 0.032	0.612 ± 0.024	0.846 ± 0.018	0.519 ± 0.022
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	0.294 ± 0.055	0.519 ± 0.087	0.763 ± 0.070	0.445 ± 0.066

Table 11: Performance of link prediction on the closed-path rule (CPR-rt) synthetic dataset with r and t symmetric (mean and standard deviation over 5 runs).

	CPR-rtu			
	h@1	h@3	h@10	MRR
TransE	0.016 ± 0.004	0.041 ± 0.009	0.101 ± 0.011	0.052 ± 0.005
RotatE	0.514 ± 0.025	0.734 ± 0.017	0.895 ± 0.012	0.645 ± 0.017
BoxE	0.420 ± 0.026	0.550 ± 0.011	0.663 ± 0.017	0.506 ± 0.018
MuRE	0.455 ± 0.023	0.608 ± 0.012	0.724 ± 0.013	0.552 ± 0.013
ComplEx	0.268 ± 0.015	0.392 ± 0.025	0.509 ± 0.025	0.354 ± 0.016
SimplE	0.282 ± 0.015	0.422 ± 0.021	0.527 ± 0.017	0.372 ± 0.016
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.491 ± 0.018	0.658 ± 0.010	0.804 ± 0.009	0.598 ± 0.010
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.003 ± 0.004	0.354 ± 0.007	0.571 ± 0.009	0.214 ± 0.005
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.494 ± 0.013	0.661 ± 0.020	0.786 ± 0.024	0.598 ± 0.010
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.529 ± 0.018	0.701 ± 0.022	0.834 ± 0.025	0.634 ± 0.018
$-\|g_{\mathbf{1};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.304 ± 0.006	0.518 ± 0.010	0.736 ± 0.013	0.447 ± 0.008
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.506 ± 0.015	0.679 ± 0.016	0.803 ± 0.023	0.610 ± 0.011
$-\|g_{\mathbf{s};\mathbf{0}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	0.474 ± 0.016	0.602 ± 0.013	0.719 ± 0.011	0.560 ± 0.014
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	0.462 ± 0.045	0.589 ± 0.025	0.711 ± 0.020	0.548 ± 0.036

Table 12: Performance of link prediction on the closed-path rule (CPR-rtu) synthetic dataset with r, t and u symmetric (mean and standard deviation over 5 runs).

	WN18RR				FB15k-237			
	h@1	h@3	h@10	MRR	h@1	h@3	h@10	MRR
Functional ExpressivE	0.407	0.519	0.619	0.482	0.256	0.387	0.535	0.350
ExpressivE	0.464	0.522	0.597	0.508	0.243	0.366	0.512	0.333
BoxE	0.400	0.472	0.541	0.451	0.238	0.374	0.538	0.337
RotatE	0.428	0.492	0.571	0.476	0.241	0.375	0.533	0.338
ComplEx	0.440	0.500	0.580	0.490	0.270	0.400	0.560	0.370
QuatE	0.436	0.499	0.572	0.482	0.271	0.401	0.556	0.366
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	0.440	0.504	0.582	0.488	0.232	0.368	0.526	0.329
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	0.459	0.517	0.579	0.500	0.228	0.356	0.514	0.322
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	0.429	0.500	0.579	0.480	0.226	0.360	0.523	0.324
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}) + b_r$	0.412	0.443	0.480	0.436	0.209	0.325	0.473	0.297

Table 13: Performance of link prediction on larger-scale datasets.

	WN18RR	FB15k-237
Functional ExpressivE	572	897
ExpressivE	5920	1839
BoxE	355	1060
RotatE	224	447
ComplEx	1102	865
QuatE	152	313
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$	159	389
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}\| + b_r$ (w/ inverse)	274	487
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}') - \mathbf{f}';\mathbf{f}\| + b_r$	407	500
$-\|g_{\mathbf{s};\mathbf{u}}(\mathbf{e};\mathbf{e}';\mathbf{e};\mathbf{e}') - \mathbf{f};\mathbf{f}';\mathbf{f}';\mathbf{f}\| + b_r$	247	580
$-d_{\mathbf{w}}(g_{\mathbf{s};\mathbf{u}}(\mathbf{e}) - \mathbf{f}) + b_r$	68	269

Table 14: Training time of different models in seconds, for the main experiments (i.e. $n = 100$ for WN18RR and $n = 200$ for FB15k-237).