Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Faithful Differentiable Reasoning with Reshuffled Region-based Embeddings

Aleksandar Pavlovic[1,2] , **Emanuel Sallinger**[1,3] , **Steven Schockaert**[4]
[1]TU Wien, Vienna, Austria
[2]Hochschule für Angewandte Wissenschaften Campus Wien, Vienna, Austria
[3]University of Oxford, Oxford, UK
[4]Cardiff University, Cardiff, UK
aleksandar.pavlovic.ai@gmail.com, emanuel.sallinger@tuwien.ac.at, schockaerts1@cardiff.ac.uk

Abstract

Knowledge graph (KG) embedding methods learn geometric representations of entities and relations to predict plausible missing knowledge. These representations are typically assumed to capture rule-like inference patterns. However, our theoretical understanding of which inference patterns can be captured remains limited. Ideally, KG embedding methods should be expressive enough such that for any set of rules, there exist relation embeddings that exactly capture these rules. This principle has been studied within the framework of region-based embeddings, but existing models are severely limited in the kinds of rule bases that can be captured. We argue that this stems from the fact that entity embeddings are only compared in a coordinate-wise fashion. As an alternative, we propose RESHUFFLE, a simple model based on ordering constraints that can faithfully capture a much larger class of rule bases than existing approaches. Most notably, RESHUFFLE can capture bounded inference w.r.t. arbitrary sets of closed path rules. The entity embeddings in our framework can be learned by a Graph Neural Network (GNN), which effectively acts as a differentiable rule base.

1 Introduction

Knowledge graph (KG) embeddings (Bordes et al. 2013; Yang et al. 2015; Trouillon et al. 2016; Sun et al. 2019) are geometric representations of knowledge graphs. Such representations are typically used to infer plausible knowledge that is not explicitly stated in the KG. An important research question is concerned with the kinds of regularities that can be captured by different kinds of models. While most standard approaches are difficult to analyse from this perspective, region-based approaches make these regularities more explicit (Gutiérrez-Basulto and Schockaert 2018; Abboud et al. 2020; Pavlovic and Sallinger 2023; Charpenay and Schockaert 2024). Essentially, in such approaches, each entity e is represented by an embedding $\mathbf{e} \in \mathbb{R}^d$ and each relation r is represented by a geometric region $Z_r \subseteq \mathbb{R}^{2d}$. The triple (e, r, f) is then captured by the embedding iff $\mathbf{e} \oplus \mathbf{f} \in Z_r$, where $\oplus$ denotes vector concatenation. In this way, we can naturally associate a KG with a given embedding. Region-based models can also associate a *rule base* with the embedding, where the rules are reflected in the spatial configuration of the regions Z_r. However, not all rule bases can be captured in this way. As a simple example, models based on TransE (Bordes et al. 2013) cannot distinguish between the rules $r_1(X_1, X_2) \wedge r_2(X_2, X_3) \rightarrow r_3(X_1, X_3)$ and $r_2(X_1, X_2) \wedge r_1(X_2, X_3) \rightarrow r_3(X_1, X_3)$. This particular limitation can be avoided by using more sophisticated region-based models (Pavlovic and Sallinger 2023; Charpenay and Schockaert 2024), but even these models can only capture particular kinds of rule bases. This appears to be related to the fact that they rely on regions which can be characterised in terms of d two-dimensional regions $Z_1^r, ..., Z_d^r$, with $Z_i^r \subseteq \mathbb{R}^2$. To check whether (e, r, f) is captured, we then check whether $(e_i, f_i) \in Z_i^r$ for each $i \in \{1, ..., d\}$, with $\mathbf{e} = (e_1, ..., e_d)$ and $\mathbf{f} = (f_1, ..., f_d)$. We will refer to such approaches as *coordinate-wise* models. Existing models primarily differ in how these two-dimensional regions are defined, e.g. ExpressivE (Pavlovic and Sallinger 2023) uses parallelograms for this purpose, while Charpenay and Schockaert (2024) used octagons.

In this paper, we propose a model that goes beyond coordinate-wise comparisons, which we term RESHUFFLE. A key challenge in designing such a model is that more flexible representations typically lead to overfitting. We avoid this problem by otherwise keeping the model as simple as possible, learning regions which are defined in terms of ordering constraints of the form $e_i \leq f_j$. As our main contribution, we show that RESHUFFLE is more expressive than existing region-based models. Furthermore, we show how entity embeddings can be learned using a Graph Neural Network (GNN) with randomly initialised node embeddings. This GNN effectively serves as a differentiable approximation of a rule base, acting on the initial representations of the entities to ensure that they capture the consequences that can be inferred from the KG. A practical consequence is that entity embeddings can thus be efficiently updated when new knowledge becomes available. From a theoretical point of view, the GNN-based formulation allows us to study bounded inference, where the number of layers of the GNN can be related to the number of inference steps.[1]

2 Related Work

Region-based Models Our theoretical understanding of the reasoning abilities of KG embedding models remains

[1]An online appendix containing all proofs and further details about our model is available at https://arxiv.org/abs/2406.09529.

poorly understood. This topic has primarily been studied in a line of work that uses region-based representations of relations (Gutiérrez-Basulto and Schockaert 2018; Abboud et al. 2020; Zhang et al. 2021; Leemhuis, Özçep, and Wolter 2022; Pavlovic and Sallinger 2023; Charpenay and Schockaert 2024). Essentially, the region-based view makes explicit which triples and rules are captured by a given embedding, which allows us to formally study what kinds of semantic dependencies a given model is capable of capturing (Gutiérrez-Basulto and Schockaert 2018; Abboud et al. 2020; Bourgaux et al. 2024). Existing work has uncovered various limitations of popular KG embedding models. For instance, Gutiérrez-Basulto and Schockaert (2018) revealed that bilinear models such as RESCAL (Nickel, Tresp, and Kriegel 2011), DistMult (Yang et al. 2015), TuckER (Balazevic, Allen, and Hospedales 2019) and ComplEx (Trouillon et al. 2016) cannot capture relation hierarchies in a faithful way. They also studied the expressivity of models that represent relations using convex polytopes, finding that arbitrary sets of closed path rules can be faithfully captured by such representations (among others). However, learning arbitrary polytopes is not feasible for high-dimensional spaces, hence more recent works have focused on finding regions that are easier to learn while still retaining some of the theoretical advantages, such as Cartesian products of boxes (Abboud et al. 2020), cones (Zhang et al. 2021; Leemhuis, Özçep, and Wolter 2022), parallelograms (Pavlovic and Sallinger 2023) and octagons (Charpenay and Schockaert 2024). However, all these models are significantly more limited in the kinds of rules that they can capture. For instance, while parallelograms and octagons makes it possible to capture closed path rules, in practice we want to capture *sets* of such rules. This is only known to be possible under rather restrictive conditions (see Section 3).

An important practical advantage of region-based models is that they enable a tight integration of symbolic rules and vector space embeddings. This makes it possible to "inject" prior knowledge in a principled way (Abboud et al. 2020) and to inspect the kinds of rules that a given model has captured. A number of embedding based approaches have been proposed with similar advantages. For instance, some methods leverage symbolic rules to regularise the embedding space (Guo et al. 2016; Tang et al. 2024). Neuro-symbolic methods which jointly learn a (differentiable approximation of) a Markov Logic Network with a KG embedding have also been proposed (Qu and Tang 2019; Chen et al. 2023). However, note that these approaches are still limited by the expressivity of the underlying KG embedding model. For instance, DiffLogic (Chen et al. 2023) aligns a Probablistic Soft Logic (Bach et al. 2017) theory with a RotatE embedding (Sun et al. 2019). RotatE, like TransE, cannot distinguish between the rules $r_1(X_1, X_2) \wedge r_2(X_2, X_3) \rightarrow r_3(X_1, X_3)$ and $r_2(X_1, X_2) \wedge r_1(X_2, X_3) \rightarrow r_3(X_1, X_3)$, hence this limitation is carried over to DiffLogic.

Inductive KG Completion Standard benchmarks for KG completion only test the reasoning abilities of models to a limited extent. In our experiments, we will therefore focus on the problem of *inductive* KG completion (Teru, Denis, and Hamilton 2020). In the inductive setting, we need to predict links between entities that are different from those that were seen during training. To perform this task, models need to learn semantic dependencies between the relations, and then exploit this knowledge when making predictions. A natural strategy is to learn rules from the training KG, either explicitly using models such as AnyBURL (Meilicke et al. 2019) and RNNLogic (Qu et al. 2021) or implicitly using differentiable rule learners such as Neural-LP (Yang, Yang, and Cohen 2017), DRUM (Sadeghian et al. 2019) and NCRL (Cheng, Ahmed, and Sun 2023).

Other approaches reduce link prediction to a graph classification problem (Teru, Denis, and Hamilton 2020). However, this requires constructing and processing a different graph for each candidate tail entity, which is inherently inefficient. NBFNet (Zhu et al. 2021) alleviates this by processing the entire graph with a single forward pass of a GNN. RED-GNN (Zhang and Yao 2022) follows a similar approach, while A*Net (Zhu et al. 2023) uses a learned heuristic to avoid processing the entire graph, providing further efficiency gains. However, in all these models, the node embeddings are query-specific, meaning that a new forward pass of the GNN is still needed for each query, which is less efficient than using KG embeddings. MorsE (Chen et al. 2022a) addresses this limitation by using a GNN to compute embeddings for previously unseen entities, and then using the embeddings for link prediction (and other tasks). We adopt a similar approach in this paper. However, in our case, each layer of the GNN essentially simulates the application of a rule base, making our method conceptually closer to differentiable rule learning methods. ReFactor GNN (Chen et al. 2022b) also uses a GNN to learn entity embeddings, by simulating the training dynamic of traditional KG embedding methods such as TransE (Bordes et al. 2013), although their method has the disadvantage that all embeddings have to be recomputed when new triples are added to the KG. Moreover, it inherits the limitations of traditional embedding models when it comes to faithfully modelling rules.

3 Problem Setting

The focus of this paper is on studying which kinds of rule bases can be faithfully captured by the proposed model. In this section, we first formally define what it means for a region-based embedding model to capture a rule base.

Preliminaries Let $\mathcal{R}$ be a set of relations, $\mathcal{E}$ a set of entities and $\mathcal{G} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ a knowledge graph. If $\mathcal{G}$ contains the triple (e, r, f) then we also say that there is an r-edge from e to f in $\mathcal{G}$. An entity embedding τ maps each entity $e \in \mathcal{E}$ to a vector $\tau(e) \in \mathbb{R}^d$. A region-based relation embedding η maps each relation $r \in \mathcal{R}$ to a geometric region $\eta(r) \subseteq \mathbb{R}^{2d}$.

Definition 1. *We say that the triple (e, r, f) is captured by the pair (τ, η), with τ an entity embedding and η a region-based relation embedding, iff $\tau(e) \oplus \tau(f) \in \eta(r)$.*

We write $\mathcal{P} \cup \mathcal{G} \models (e, r, f)$ to denote that the triple (e, r, f) can be entailed from the rule base $\mathcal{P}$ and the knowledge graph $\mathcal{G}$. More precisely, we have $\mathcal{P} \cup \mathcal{G} \models (e, r, f)$ iff either $(e, r, f) \in \mathcal{G}$, or $\mathcal{P}$ contains a rule $r_1(X_1, X_2) \wedge r_2(X_2, X_3) \wedge ... \wedge r_p(X_p, X_{p+1}) \rightarrow r(X_1, X_{p+1})$ such that $\mathcal{P} \cup \mathcal{G} \models (e, r_1, e_2)$, $\mathcal{P} \cup \mathcal{G} \models (e_2, r_2, e_3)$, ..., $\mathcal{P} \cup \mathcal{G} \models (e_p, r_p, f)$ for some entities $e_2, ..., e_p$. We furthermore write $\mathcal{P} \models \rho$, for a rule ρ, to denote that $\mathcal{P}$ entails ρ w.r.t. the standard notion of entailment from first-order logic (when viewing rules as universally quantified material implications).

Capturing Closed Path Rules Similar to most existing rule-based methods for KG completion (Yang, Yang, and Cohen 2017; Meilicke et al. 2019; Sadeghian et al. 2019; Qu et al. 2021; Cheng, Ahmed, and Sun 2023), we focus on *closed path rules*, which are rules ρ of the form:

$$r_1(X_1, X_2) \wedge r_2(X_2, X_3) \wedge \\ ... \wedge r_p(X_p, X_{p+1}) \rightarrow r(X_1, X_{p+1}) \quad (1)$$

We refer to $r_1(X_1, X_2) \wedge r_2(X_2, X_3) \wedge ... \wedge r_p(X_p, X_{p+1})$ as the body of the rule and to $r(X_1, X_{p+1})$ as the head.

Definition 2. *We say that a region-based relation embedding η captures a rule of the form* (1) *if for all vectors $\mathbf{x_1}, ..., \mathbf{x_{p+1}} \in \mathbb{R}^d$ such that that $(\mathbf{x_1} \oplus \mathbf{x_2} \in \eta(r_1)) \wedge .... \wedge (\mathbf{x_p} \oplus \mathbf{x_{p+1}} \in \eta(r_p))$ we also have $(\mathbf{x_1} \oplus \mathbf{x_{p+1}} \in \eta(r))$.*

Apart from their practical significance, our focus on closed path rules is also motivated by the observation that existing region-based models have particular limitations when it comes to capturing this kind of rules. Some approaches, such as BoxE (Abboud et al. 2020) are not capable of capturing such rules at all. More recent approaches (Pavlovic and Sallinger 2023; Charpenay and Schockaert 2024) are capable of capturing closed path rules, but with significant limitations when it comes to modelling sets of such rules.

Eq-complete Knowledge Graphs Throughout this paper, we will assume that all knowledge graphs $\mathcal{G}$ contain the triple (e, eq, e) for every entity e. We will refer to such knowledge graphs as *eq*-complete. This assumption is similar to the common practice of adding self-loops in GNN models. In our setting, it will mean that instead of directly capturing a rule like $r_1(X_1, X_2) \wedge r_2(X_2, X_3) \rightarrow r(X_1, X_3)$ we might instead capture a rule like $r_1(X_1, X_2) \wedge eq(X_2, X_3) \wedge r_2(X_3, X_4) \rightarrow r(X_1, X_4)$. This latter rule is equivalent, in the sense that any triple that can be inferred from an *eq*-complete KG with the former rule, can also be inferred with the latter rule. This offers some modelling flexibility which will be important in our approach.

Faithfully Capturing Rule Bases Definition 2 specifies what it means for a closed path rule to be captured. Our main research question is whether it is possible to faithfully capture a *set* of closed path rules $\mathcal{P}$. In other words, can parameters be found for the matrices $\mathbf{B_r}$ such that *all rules* entailed by $\mathcal{P}$ are captured, and *only those rules*. This is made precise in the following definition.[2]

[2]Our notion of faithfully capturing rule bases is closely related to the notion of exactly and exclusively capturing a language of patterns, from Abboud et al. (2020), and the notion of strong TBox-faithfulness from Bourgaux et al. (2024). It is, however, slightly weaker due to restriction to *eq*-complete knowledge graphs.

Definition 3. *We say that a region-based relation embedding η faithfully captures the rule base $\mathcal{P}$ if for every eq-complete knowledge graph $\mathcal{G}$, the following conditions hold:*

1. *Suppose that $\mathcal{P} \cup \mathcal{G} \models (a, r, b)$ and let τ be an entity embedding such that (τ, η) captures every triple in $\mathcal{G}$. Then (τ, η) captures the triple (a, r, b) as well.*
2. *Suppose that $\mathcal{P} \cup \mathcal{G} \not\models (a, r, b)$. There exists an entity embedding τ such that (τ, η) captures every triple in $\mathcal{G}$ but not the triple (a, r, b).*

Existing models are only able to faithfully capture sets of closed path rules in specific cases. For instance, Charpenay and Schockaert (2024) showed this to be possible when every non-trivial rule entailed from $\mathcal{P}$ is a closed path rule of the form (1) in which $r_1, ..., r_p, r$ are all distinct relations. For instance, rules of the form $r_1(X_1, X_2) \wedge r_1(X_2, X_3) \rightarrow r(X_1, X_3)$ are not covered by their result. Similarly, they cannot capture rule bases with cyclic dependencies such as:

$$\mathcal{P} = \{r_1(X_1, X_2) \wedge r_2(X_2, X_3) \rightarrow r_3(X_1, X_3), \\ r_3(X_1, X_2) \wedge r_4(X_2, X_3) \rightarrow r_1(X_1, X_3)\}$$

Note that while we focus our theoretical analysis on whether a given rule base $\mathcal{P}$ can be captured, in practice we normally do not have access to such a rule base. We study whether our model is capable of capturing rule bases because this is a necessary condition to allow it to *learn* semantic dependencies in the form of rules.

4 Model Formulation

Our aim is to introduce a knowledge graph embedding model which goes beyond coordinate-wise models, but which otherwise remains as simple as possible. The central idea is to define the regions $\eta(r)$ using coordinate-wise ordering constraints between reshuffled entity embeddings:

$$\eta(r) = \{(e_1, ..., e_d, f_1, ..., f_d) \,|\, \forall i \in I_r \,.\, e_{\sigma_r(i)} \leq f_i\} \quad (2)$$

where the representation of a region r is parameterised by a set of coordinates $I_r \subseteq \{1, ..., d\}$ and a mapping $\sigma_r : I_r \rightarrow \{1, ..., d\}$. We thus need a maximum of $2d$ parameters to completely specify the embedding of a given relation. Note that in the special case where $I_r = \emptyset$, we have $\eta(r) = \mathbb{R}^{2d}$.

Example 1. *Let $\mathbf{e} = (0, 0, 0)$, $\mathbf{f} = (0, 0, 1)$ and $\mathbf{g} = (2, 2, 0)$ be the embeddings of entities e, f, g. Let the relations r_1, r_2, r_3 be represented as follows: $I_{r_1} = \{3\}$, $I_{r_2} = \{1, 2\}$, $I_{r_3} = \{1\}$, $\sigma_{r_1}(3) = 2$, $\sigma_{r_2}(1) = \sigma_{r_2}(2) = 3$ and $\sigma_{r_3}(1) = 2$. Then we find that $\mathbf{e} \oplus \mathbf{f} \in \eta(r_1)$, meaning that the triple (e, r_1, f) is captured. Indeed, for $\mathbf{e} \oplus \mathbf{f} \in \eta(r_1)$ to hold, we need $e_{\sigma_{r_1}(3)} = e_2 \leq f_3$, which is satisfied. We similarly find that (f, r_2, g) is captured, because $f_{\sigma_{r_2}(1)} = f_3 \leq g_1$ and $f_{\sigma_{r_2}(2)} = f_3 \leq g_2$.*

The following example illustrates how the use of ordering constraints allows us to capture rules.

Example 2. *Consider a rule $r_1(X, Y) \wedge r_2(Y, Z) \rightarrow r_3(X, Z)$. This rule is captured by an embedding of the form* (2) *if for each $i \in I_{r_3}$ we have that $i \in I_{r_2}$, $\sigma_{r_2}(i) \in I_{r_1}$ and $\sigma_{r_1}(\sigma_{r_2}(i)) = \sigma_{r_3}(i)$. For instance, the relations r_1, r_2, r_3 from Example 1 satisfy these conditions.*

In general, if these conditions are satisfied and the triples (e, r_1, f) *and* (f, r_2, g) *are captured, then for each* $i \in I_{r_3}$ *we have:* $e_{\sigma_{r_1}(\sigma_{r_2}(i))} \leq f_{\sigma_{r_2}(i)} \leq g_i$*. Since we assumed* $\sigma_{r_1}(\sigma_{r_2}(i)) = \sigma_{r_3}(i)$ *it follows that* $e_{\sigma_{r_3}(i)} \leq g_i$ *for every* $i \in I_r$ *and thus that the triple* (e, r_3, f) *is also captured.*

Characterising Ordering Constraints We now turn our focus to how the proposed model can be represented in a way that enables efficient GPU computations. Note that we can characterise $\eta(r)$ as follows:

$$\eta(r) = \{\mathbf{e} \oplus \mathbf{f} \mid \max(\mathbf{A_r e}, \mathbf{f}) = \mathbf{f}\} \tag{3}$$

where the maximum is applied component-wise and the matrix $\mathbf{A_r} \in \mathbb{R}^{d \times d}$ is constrained such that (i) all components are either 0 or 1 and (ii) at most one component in each row is non-zero. As we explain below, the entity embeddings will be constructed using a GNN, starting from randomly initialised embeddings. For this approach to be effective, the dimensionality of the entity embeddings needs to be sufficiently high, to prevent randomly initialised embeddings from capturing KG triples by chance. At the same time, the number of parameters should remain sufficiently low to prevent overfitting. For this reason, we decouple the number of parameters from the dimensionality of the embeddings. To this end, we learn matrices $\mathbf{A_r}$ of the following form:

$$\mathbf{A_r} = \mathbf{B_r} \otimes \mathbf{I_k} \tag{4}$$

where we write $\otimes$ for the Kronecker product, $\mathbf{I_k}$ is the k-dimensional identity matrix and $\mathbf{B_r}$ is an $\ell \times \ell$ matrix, with $d = k\ell$. The rows of $\mathbf{B_r}$ are constrained similarly as those of $\mathbf{A_r}$, i.e. each row is either a one-hot vector or a 0-vector. To make the computations more efficient, we represent each entity using a matrix $\mathbf{Z_e} \in \mathbb{R}^{\ell \times k}$, rather than a vector, and define the region-based embeddings as follows:

$$\eta(r) = \{\mathit{flatten}(\mathbf{Z_e}) \oplus \mathit{flatten}(\mathbf{Z_f}) \mid \max(\mathbf{B_r Z_e}, \mathbf{Z_f}) = \mathbf{Z_f}\} \tag{5}$$

where we write *flatten*$(\mathbf{Z_e})$ for the vector that is obtained by concatenating the rows of $\mathbf{Z_e}$. We will refer to the region-based embedding model defined in (5) as RESHUFFLE. Note that a triple (e, r, f) is captured if $\mathbf{B_r Z_e} \preceq \mathbf{Z_f}$, where $\mathbf{X} \preceq \mathbf{Y}$ iff $\max(\mathbf{X}, \mathbf{Y}) = \mathbf{Y}$. It is furthermore easy to verify that a rule of the form (1) is satisfied if $\mathbf{B_r} \preceq \mathbf{B_{r_p}} \mathbf{B_{r_{p-1}}} \cdots \mathbf{B_{r_1}}$.

Learning Entity Embeddings with GNNs The format of (5) suggests how entity embeddings in our framework can be learned using a GNN. A practical advantage of using a GNN for this purpose is that we can use our model for inductive KG completion. As we will see in Section 6.2, the use of a GNN also has an important theoretical advantage, as it allows us to capture bounded reasoning.

Let us write $\mathbf{Z_e^{(l)}} \in \mathbb{R}^{\ell \times k}$ for the representation of entity e in layer l of the GNN. Starting from (5), we naturally end up with the following message-passing GNN:

$$\mathbf{Z_f^{(l+1)}} = \max\left(\{\mathbf{Z_f^{(l)}}\} \cup \{\mathbf{B_r Z_e^{(l)}} \mid (e, r, f) \in \mathcal{G}\}\right) \tag{6}$$

The embeddings $\mathbf{Z_e^{(0)}}$ are initialised randomly, such that (i) all coordinates are non-negative, (ii) the coordinates of different entity embeddings are sampled independently, and (iii) there are at least two distinct values that have a non-negative probability of being sampled for each coordinate. In this way, the probability of a constraint $\mathbf{B_r Z_e^{(0)}} \preceq \mathbf{Z_f^{(0)}}$ being satisfied by chance can be made arbitrarily small, without increasing the number of parameters of the model, by choosing the value of k to be sufficiently high.

5 Constructing Models from Rule Graphs

For any knowledge graph $\mathcal{G}$, the GNN defined in (6) can be used to construct an entity embedding τ such that (τ, η) captures every triple from $\mathcal{G}$. To see this, first note that because of the use of the maximum, the GNN always converges after a finite number of iterations. Upon convergence, it is clear that every triple of $\mathcal{G}$ must indeed be satisfied. However, the resulting entity embeddings may also capture triples which are not in $\mathcal{G}$. Our central research question is whether we can always choose the matrices $\mathbf{B_r}$ such that a triple is captured by these entity embeddings iff the triple can be inferred from $\mathcal{G} \cup \mathcal{P}$, for a given rule base $\mathcal{P}$. In other words, given a set of closed path rules $\mathcal{P}$, can we always construct a RESHUFFLE model that faithfully captures it? Rather than constructing the matrices $\mathbf{B_r}$ directly, we first introduce the notion of a rule graph, which will serve as a convenient abstraction.

Rule Graphs Let $\mathcal{P}$ be a set of closed path rules. We associate with $\mathcal{P}$ a labelled multi-graph $\mathcal{H}$, i.e. a set of triples (n_1, r, n_2). Note that this graph is formally equivalent to a knowledge graph, but the nodes in this case do not correspond to entities. Rather, as we will see, they correspond to the different rows/columns of the matrices $\mathbf{B_r}$. A path in $\mathcal{H}$ from n_1 to n_{p+1} is a sequence of triples of the form $(n_1, r_1, n_2), (n_2, r_2, n_3), ..., (n_p, r_p, n_{p+1})$. The *type* of this path is given by the sequence of relations $r_1; r_2; ...; r_p$. The *eq*-reduced type of the path is obtained by removing all occurrences *eq* in $r_1; r_2; ...; r_p$. For instance, for a path of type r_1; *eq*; *eq*; r_2; *eq*, the *eq*-reduced type is $r_1; r_2$.

Definition 4. *Let* $\mathcal{R}$ *be a set of relations, and let* $\mathcal{P}$ *be a set of rules defined over these relations. A rule graph* $\mathcal{H}$ *for* $\mathcal{P}$ *and* $\mathcal{R}$ *is a labelled multi-graph, where the labels are taken from* $\mathcal{R}$*, such that the following properties are satisfied:*

(R1) *For every relation* $r \in \mathcal{R}$*, there is some edge in* $\mathcal{H}$ *labelled with* r*.*

(R2) *For every node* n *in* $\mathcal{H}$ *and every* $r \in \mathcal{R}$*, it holds that* n *has at most one incoming* r*-edge.*

(R3) *Suppose there is an* r*-edge in* $\mathcal{H}$ *from node* n_1 *to node* n_2*. Suppose furthermore that* $\mathcal{P} \models r_1(X_1, X_2) \wedge r_2(X_2, X_3) \wedge ... \wedge r_p(X_p, X_{p+1}) \rightarrow r(X_1, X_{p+1})$*. Then there is a path in* $\mathcal{H}$ *from* n_1 *to* n_2 *whose eq-reduced type is* $r_1; ...; r_p$*.*

(R4) *Suppose that for every* r*-edge, there is a path connecting the same nodes whose eq-reduced type belongs to* $\{(r_{11}; ...; r_{1p_1}), ..., (r_{q1}; ...; r_{qp_q})\}$*. Then there is some* $i \in \{1, ..., q\}$ *such that that* $\mathcal{P} \models r_{i1}(X_1, X_2) \wedge ... \wedge r_{ip_i}(X_{p_i}, X_{p_{i+1}}) \rightarrow r(X_1, X_{p_{i+1}})$*.*

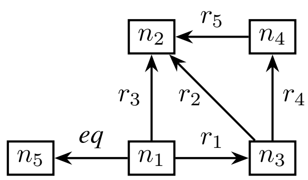

Figure 1: Rule graph for $\mathcal{P}_1$.

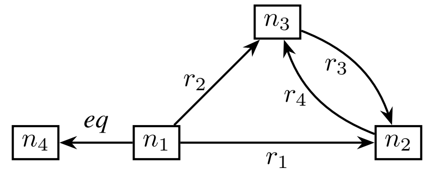

Figure 2: Rule graph for $\mathcal{P}_2$.

When $\mathcal{R}$ is clear from the context, we will simply refer to $\mathcal{H}$ as a rule graph for $\mathcal{P}$. The definition reflects the fact that a rule is captured when the ordering constraints associated with its body entail the ordering constraints associated with its head, as was illustrated in Example 2. Specifically, this is encoded by condition (R3). Condition (R4) is needed to ensure that *only* the rules in $\mathcal{P}$ are captured. Note that we require (R4) to be satisfied for *every* set of path types $\{(r_{11};...;r_{1p_1}),...,(r_{q1};...;r_{qp_q})\}$ that satisfy the condition. The following example illustrates why (R4) is needed.

Example 3. *Suppose there are two r-edges in the rule graph, namely between n_1 and n_2 and between n_3 and n_4. Suppose furthermore that there is an $r_1;r_2$ path connecting n_1 and n_2, and an $r_3;r_4$ path connecting n_3 and n_4. Suppose that the knowledge graph $\mathcal{G}$ contains the triples $(a,r_1,b),(b,r_2,c),(a,r_3,d),(d,r_4,c)$. Then, as we will see, the corresponding model would capture the triple (a,r,c). In other words, the model would capture the rule $r_1(X_1,X_2)\wedge r_2(X_2,X_3)\wedge r_3(X_1,X_4)\wedge r_4(X_4,X_3)\rightarrow r(X_1,X_3)$, but such a rule can never be entailed from $\mathcal{P}$, since the latter only contains closed path rules. We can thus only allow such rule graphs if $r_1(X_1,X_2)\wedge r_2(X_2,X_3)\rightarrow r(X_1,X_3)$ or $r_3(X_1,X_4)\wedge r_4(X_4,X_3)\rightarrow r(X_1,X_3)$ is entailed from $\mathcal{P}$, which is what (R4) ensures.*

Conditions (R1) and (R2) are needed because, in the construction we consider below, the nodes of the rule graph will correspond to the rows/columns of the matrices $\mathbf{B_r}$. Condition (R1) ensures that $\mathbf{B_r}$ contains at least one non-zero component for each relation r, while (R2) ensures that each row of $\mathbf{B_r}$ has at most one non-zero component.

Example 4. *Let $\mathcal{P}_1$ consist of the following rules:*

$$r_1(X,Y)\wedge r_2(Y,Z)\rightarrow r_3(X,Z)$$
$$r_4(X,Y)\wedge r_5(Y,Z)\rightarrow r_2(X,Z)$$

A corresponding rule graph is shown in Figure 1. As an example with cyclic dependencies, let $\mathcal{P}_2$ consist of:

$$r_2(X,Y)\wedge r_3(Y,Z)\rightarrow r_1(Y,Z)$$
$$r_1(X,Y)\wedge r_4(Y,Z)\rightarrow r_2(X,Z)$$

A corresponding rule graph is shown in Figure 2.

Constructing Models Given a rule graph $\mathcal{H}$, we define the corresponding parameters of a ReshufflE model (i.e. the matrices $\mathbf{B_r}$) as follows. Each node from the rule graph is associated with one row/column of $\mathbf{B_r}$. Let $n_1,...,n_\ell$ be an enumeration of the nodes in the rule graph. The corresponding matrix $\mathbf{B_r}=(b_{ij})$ is defined as:

$$b_{ij}=\begin{cases}1 & \text{if } \mathcal{H} \text{ has an } r\text{-edge from } n_j \text{ to } n_i\\ 0 & \text{otherwise}\end{cases} \tag{7}$$

Note that because of condition (R2), there will be at most one non-zero element in each row of $\mathbf{B_r}$, in accordance with the assumptions that we made in Section 4.

If a set of closed path rules $\mathcal{P}$ has a rule graph $\mathcal{H}$ then the corresponding ReshufflE model, defined as in (7), faithfully captures $\mathcal{P}$. To prove this, we need to show that the two conditions from Definition 3 are satisfied. The following proposition shows that the first condition is satisfied.

Proposition 1. *Let $\mathcal{P}$ be a set of closed path rules and $\mathcal{G}$ an eq-complete knowledge graph. Suppose $\mathcal{P}\cup\mathcal{G}\models(a,r,b)$. Let $\mathcal{H}$ be a rule graph for $\mathcal{P}$ and let η be the corresponding* ReshufflE *model. Let τ be an entity embedding such that (τ,η) captures every triple in $\mathcal{G}$. It holds that (τ,η) captures the triple (a,r,b).*

To show that the second condition of Definition 3 is also satisfied, we need to show the existence of a particular entity embedding. We show that the embedding constructed by the GNN in (6) satisfies this condition with a probability that can be made arbitrarily high. The reason why this embedding is not guaranteed to satisfy the condition is because, for any triple (e,r,f), there is always a chance that it is captured by the model, even if $\mathcal{P}\cup\mathcal{G}\not\models(e,r,f)$, due to the fact that the entity embeddings are initialised randomly. However, by choosing the dimensionality of the entity embeddings to be sufficiently large, we can make the probability of this happening arbitrarily small. As before, we write ℓ to denote the number of rows in $\mathbf{Z}_e$ and k for the number of columns. Note that the value of k does not affect the number of parameters, since the size of the matrices $\mathbf{B_r}$ only depends on ℓ and the entity embeddings are randomly initialised. In practice, we can thus simply choose k to be sufficiently large.

Proposition 2. *Let $\mathcal{P}$ be a set of closed path rules and $\mathcal{G}$ an eq-complete knowledge graph. Let $\mathcal{H}$ be a rule graph for $\mathcal{P}$ and let $\mathbf{Z_e^{(1)}}$ be the entity representations learned using the GNN* (6) *for the corresponding* ReshufflE *model. For any $\varepsilon>0$, there exists some $k_0\in\mathbb{N}$ such that, when $k\geq k_0$, for any $m\in\mathbb{N}$ and $(a,r,b)\in\mathcal{E}\times\mathcal{R}\times\mathcal{E}$ such that $\mathcal{P}\cup\mathcal{G}\not\models(a,r,b)$, we have*

$$Pr[\mathbf{B_r}\mathbf{Z_a^{(m)}}\preceq\mathbf{Z_b^{(m)}}]\leq\varepsilon$$

Finally, there always exists an initialisation of the entity embeddings for which the embeddings learned by the GNN satisfy the second condition of Definition 3, even when $k=1$. In particular, we have the following result

Proposition 3. *Let $\mathcal{P}$ be a set of closed path rules. Let $\mathcal{H}$ be a rule graph for $\mathcal{P}$ and let η be the corresponding* RESHUFFLE *model. It holds that η faithfully captures $\mathcal{P}$.*

6 Constructing Rule Graphs

We now return to the central question of this paper: given a set of closed path rules $\mathcal{P}$, is it possible to construct a RESHUFFLE model which faithfully captures $\mathcal{P}$? Thanks to Proposition 3 we know that this is the case when a rule graph for $\mathcal{P}$ exists. The key question thus becomes whether it is always possible to construct such a rule graph. As the following result shows, if there are no cyclic dependencies in $\mathcal{P}$, a rule graph always exists.

Proposition 4. *Let $\mathcal{P}$ be a rule base. Assume that we can rank the relations in $\mathcal{R}$ as $r_1, ..., r_{|\mathcal{R}|}$, such that for every rule in $\mathcal{P}$ with r_i in the body and r_j in the head, it holds that $i < j$. There exists a rule graph for $\mathcal{P}$.*

It follows that the class of rule bases that can be captured by RESHUFFLE models is strictly larger than the class that has been considered in previous work (Charpenay and Schockaert 2024). Unfortunately, there exist rule bases with cyclic dependencies for which no valid rule graph can be found.

Example 5. *Let $\mathcal{P}$ contain the following rule:*

$$r_1(X, Y) \wedge r_2(Y, Z) \wedge r_1(Z, U) \rightarrow r_2(X, U)$$

To see why there is no rule graph for $\mathcal{P}$, consider the following knowledge graph $\mathcal{G}$:

$$\mathcal{G}=\{(x_1, r_1, x_2), (x_2, r_1, x_3), ..., (x_{l-1}, r_1, r_l), \\ (x_l, r_2, x_{l+1}), (x_{l+1}, r_1, x_{l+2}), ..., (x_k, r_1, x_{k+1})\}$$

We have that $\mathcal{P} \cup \mathcal{G} \models (x_1, r_2, x_{k+1})$ only if the number of repetitions of r_1 at the start of the sequence matches the number at the end, but rule graphs cannot encode this.

The argument from the previous example can be formalised as follows. Let $\mathcal{P}$ be a set of closed path rules. Let $\mathcal{R}_1$ be the set of relations from $\mathcal{R}$ that appear in the head of some rule in $\mathcal{P}$. For any $r \in \mathcal{R}_1$, we can consider a context-free grammar with two types of production rules:

- For each rule $r_1(X_1, X_2) \wedge ... \wedge r_p(X_p, X_{p+1}) \rightarrow r(X_1, X_{p+1})$, there is a production rule $r \Rightarrow r_1 r_2 ... r_p$.
- For each $r \in \mathcal{R}_1$, there is a production rule $r \Rightarrow \overline{r}$.

The elements of $(\mathcal{R} \setminus \mathcal{R}_1) \cup \{\overline{r} \mid r \in \mathcal{R}_1\}$ are terminal symbols, those in $\mathcal{R}_1$ are non-terminal symbols, and r is the start symbol. We write L_r for the corresponding language.

Proposition 5. *Let $\mathcal{P}$ be a set of closed path rules and suppose that there exists a rule graph $\mathcal{H}$ for $\mathcal{P}$. Let $\mathcal{R}_1$ be the set of relations that appear in the head of some rule in $\mathcal{P}$. It holds that the language L_r is regular for every $r \in \mathcal{R}_1$.*

This shows that we cannot capture arbitrary rule bases using rule graphs. For instance, for the rule base from Example 5, we have $L_{r_2} = \{r_1^l \overline{r}_2 r_1^l \mid l \in \mathbb{N} \setminus \{0\}\}$, where we write x^l for the string that consists of l repetitions of x. It is well-known that this language is not regular.

Following this negative result, we now establish two important positive results. First, in Section 6.1, inspired by regular grammars, we introduce a special class of rule bases with cyclic dependencies for which a rule graph is guaranteed to exist. Second, in Section 6.2, we focus on the practically important setting of bounded inference: since GNNs use a fixed number of layers in practice, what mostly matters is what can be derived in a bounded number of steps. It turns out that if we only care about such inferences, we can capture arbitrary sets of closed path rules.

6.1 Left-Regular Rule Bases

To show that many rule bases with cyclic dependencies can still be captured, we consider the following notion of a left-regular rule base, inspired by left-regular grammars.

Definition 5. *Let $\mathcal{P}$ be a rule base. Let $\mathcal{R}_1$ be the set of relations that appear in the head of a rule from $\mathcal{P}$. We call $\mathcal{P}$ left-regular if every rule is of the following form:*

$$r_1(X, Y) \wedge r_2(Y, Z) \rightarrow r_3(X, Z) \quad (8)$$

such that $r_2 \notin \mathcal{R}_1$.

While Definition 5 only considers rules with two relations in the body, rules with more than two atoms can straightforwardly be simulated by introducing fresh relations. The following result shows that left-regular rule bases can always be faithfully captured by a RESHUFFLE model.

Proposition 6. *For any left-regular set of closed path rules $\mathcal{P}$, there exists a rule graph for $\mathcal{P}$.*

Proof. (Sketch) Given a left-regular rule base $\mathcal{P}$, we construct the corresponding rule graph $\mathcal{H}$ as follows.

1. We add the node n_0.
2. For each relation $r \in \mathcal{R}$, we add a node n_r, and we connect n_0 to n_r with an r-edge.
3. For each rule of the form (8), we add an r_2-edge from n_{r_1} to n_{r_3}.
4. For each node n with multiple incoming r-edges for some $r \in \mathcal{R}$, we do the following. Let $\sharp(r, n)$ be the number of incoming r-edges for node n. Let $p = \max_{r \in \mathcal{R}} \sharp(r, n)$. We create fresh nodes $n_1, ..., n_{p-1}$ and add *eq*-edges from n_i to n_{i-1} ($i \in \{1, ..., p-1\}$), where we define $n_0 = n$. Let $r \in \mathcal{R}$ be such that $\sharp(r, n) > 1$. Let $n'_0, ..., n'_q$ be the nodes with an r-link to n; then we have $q \leq p - 1$. For each $i \in \{1, ..., q\}$ we replace the edge from n'_i to n by an edge from n'_i to n_i.

The correctness of this process is shown in the online appendix. $\square$

6.2 Bounded Inference

In practice, the GNN can only carry out a finite number of inference steps. Rather than requiring that the resulting embeddings capture all triples that can be inferred from $\mathcal{P} \cup \mathcal{G}$, it is thus natural to merely require that the embeddings capture all triples that can be inferred using a bounded number of inference steps. We know from Proposition 5 that it is not always possible to construct a rule graph for a given rule base $\mathcal{P}$. To address this, we now weaken the notion of a rule graph, aiming to capture reasoning up to a fixed number of inference steps. In the following, we will assume that $\mathcal{P}$ only contains rules with two relations in the body. Note that we can assume this w.l.o.g. as any set of closed path rules can be converted into such a format by introducing fresh relations.

Let us write $\mathcal{P}\cup\mathcal{G} \models_m (e, r, f)$ to denote that (e, r, f) can be derived from $\mathcal{P} \cup \mathcal{G}$ in m steps. More precisely, we have $\mathcal{P} \cup \mathcal{G} \models_0 (e, r, f)$ iff $(e, r, f) \in \mathcal{G}$. Furthermore, we have $\mathcal{P} \cup \mathcal{G} \models_m (e, r, f)$, for $m > 0$, iff $\mathcal{P} \cup \mathcal{G} \models_{m-1} (e, r, f)$ or there is a rule $r_1(X_1, X_2) \wedge r_2(X_2, X_3) \rightarrow r(X_1, X_3)$ in $\mathcal{P}$ and an entity $g \in \mathcal{E}$ such that $\mathcal{P} \cup \mathcal{G} \models_{m_1} (e, r_1, g)$ and $\mathcal{P} \cup \mathcal{G} \models_{m_2} (g, r_2, f)$, with $m = m_1 + m_2 + 1$.

Definition 6. *Let $m \in \mathbb{N}$. We call $\mathcal{H}$ an m-*bounded rule graph *for $\mathcal{P}$ if $\mathcal{H}$ satisfies conditions (R1)–(R3) as well as the following weakening of (R4):*

(R4m) *Suppose that for every r-edge, there is a path connecting the same nodes whose eq-reduced type belongs to $\{(r_{11}; ...; r_{1p_1}), ..., (r_{q1}; ...; r_{qp_q})\}$, with $p_1, ..., p_q \leq m + 1$. Then there is some $i \in \{1, ..., q\}$ such that that $\mathcal{P} \models_m r_{i1}(X_1, X_2) \wedge ... \wedge r_{ip_i}(X_{p_i}, X_{p_{i+1}}) \rightarrow r(X_1, X_{p_{i+1}})$.*

Given an m-bounded rule graph, we can again construct a corresponding RESHUFFLE model using (7). Moreover, Proposition 1 remains valid for m-bounded rule graphs. Proposition 2 can be weakened as follows.

Proposition 7. *Let $\mathcal{P}$ be a set of closed path rules $\mathcal{G}$ an eq-complete knowledge graph. Let $\mathcal{H}$ be an m-bounded rule graph for $\mathcal{P}$ and let $\mathbf{Z}_\mathbf{e}^{(\mathbf{l})}$ be the entity representations that are learned by the GNN, for the corresponding* RESHUFFLE *model. For any $\varepsilon > 0$, there exists some $k_0 \in \mathbb{N}$ such that, when $k \geq k_0$, for any $i \leq m + 1$ and $(a, r, b) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ such that $\mathcal{P} \cup \mathcal{G} \not\models_m (a, r, b)$, we have*

$$Pr[\mathbf{B_r Z_a^{(i)}} \preceq \mathbf{Z_b^{(i)}}] \leq \varepsilon$$

We can again show that there always exists an initialisation of the entity embeddings such that the triple (a, r, b) is not captured by the resulting entity embeddings; we omit the details. Crucially, we have the following result, showing that m-bounded rule graphs always exist, and hence that RESHUFFLE models can correctly capture bounded reasoning for arbitrary sets of closed path rules.

Proposition 8. *For any set of closed path rules $\mathcal{P}$, there exists an m-bounded rule graph for $\mathcal{P}$.*

Proof. (Sketch) Given a set of closed path rules $\mathcal{P}$ we can construct an m-bounded rule graph as follows.

1. We add the node n_0.
2. For each relation $r \in \mathcal{R}$, we add a node n_r, and we connect n_0 to n_r with an r-edge.
3. We repeat the following until convergence. Let $r \in \mathcal{R}$ and assume there is an r-edge from n to n'. Let $r_1(X, Y) \wedge r_2(Y, Z) \rightarrow r(X, Z)$ be a rule from $\mathcal{P}$ and suppose that there is no $r_1; r_2$ path connecting n and n'. Suppose furthermore that the edge (n, n') is on some path from n_0 to a node $n_{r'}$, with $r' \in \mathcal{R}$, whose length is at most m. We add a fresh node n'' to the rule graph, an r_1-edge from n to n'', and an r_2-edge from n'' to n'.
4. For each $r \in \mathcal{R}$ and r-edge (n, n') such that for some rule $r_1(X, Y) \wedge r_2(Y, Z) \rightarrow r(X, Z)$ from $\mathcal{P}$ there is no $r_1; r_2$ path connecting n and n', we do the following:
   (a) We add a fresh node n'', an r_1-edge from n to n'' and an r_2-edge from n'' to n'.
   (b) We repeat the following until convergence. For each r'-edge from n to n'' and each rule $r'_1(X, Y) \wedge r'_2(Y, Z) \rightarrow r'(X, Z)$ from $\mathcal{P}$, we add an r'_1 edge from n to n'' and an r'_2-loop to n'' (if no such edges/loops exist yet).
   (c) We repeat the following until convergence. For each r'-edge from n'' to n' and each rule $r'_1(X, Y) \wedge r'_2(Y, Z) \rightarrow r'(X, Z)$ from $\mathcal{P}$, we add an r'_1-loop to n'' and an r'_2-edge from n'' to n' (if no such edges/loops exist yet).
   (d) We repeat the following until convergence. For each r'-loop at n'', and each rule $r'_1(X, Y) \wedge r'_2(Y, Z) \rightarrow r'(X, Z)$ from $\mathcal{P}$, we add an r'_1-loop and an r'_2-loop to n'' (if no such loops exist yet).
5. For each node n with multiple incoming r-edges for one or more relations from $\mathcal{R}$, we do the following. Let $\sharp(r, n)$ be the number of incoming r-edges for node n. Let $p = \max_{r\in\mathcal{R}} \sharp(r, n)$. We create fresh nodes $n_1, ..., n_{p-1}$ and add *eq*-edges from n_i to n_{i-1} ($i \in \{1, ..., p - 1\}$), where we define $n_0 = n$. Let $r \in \mathcal{R}$ be such that $\sharp(r, n) > 1$. Let $n'_0, ..., n'_q$ be the nodes with an r-link to n; then we have $q \leq p - 1$. For each $i \in \{1, ..., q\}$ we replace the edge from n'_i to n by an edge from n'_i to n_i.

The correctness of this process is shown in the online appendix. □

7 Beyond Closed Path Rules

Thus far we have only focused on sets of closed path rules, motivated by their importance for KG completion, and the known limitations of existing region-based models when it comes to capturing such rules. Two other important types of rules are hierarchy and intersection rules, which are respectively of the following form:

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

Existing models (Abboud et al. 2020; Pavlovic and Sallinger 2023; Charpenay and Schockaert 2024) are capable of capturing arbitrary sets of such rules. We now show that the same is true for RESHUFFLE (with high probability).

Proposition 9. *Let $\mathcal{P}$ be a set of hierarchy and intersection rules. There exists a* RESHUFFLE *model such that for every knowledge graph $\mathcal{G}$ the following conditions are satisfied:*

1. *Suppose that $\mathcal{P} \cup \mathcal{G} \models (a, r, b)$ and let τ be an entity embedding such that (τ, η) captures every triple in $\mathcal{G}$. It holds that (τ, η) captures (a, r, b).*
2. *For any $\varepsilon > 0$ there is a $k_0 \in \mathbb{N}$ such that, when $k \geq k_0$, for any $m \in \mathbb{N}$ and $(a, r, b) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ such that $\mathcal{P} \cup \mathcal{G} \not\models (a, r, b)$, we have $Pr[\mathbf{B_r Z_a^{(m)}} \preceq \mathbf{Z_b^{(m)}}] \leq \varepsilon$, where $\mathbf{Z_e^{(m)}}$ are the entity representations that are learned by the GNN* (6).

		FB15k-237				WN18RR				NELL-995			
		v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
GNN	CoMPILE	0.676	0.829	0.846	0.874	0.836	0.798	0.606	0.754	0.583	0.938	0.927	0.751
	GraIL	0.642	0.818	0.828	0.893	0.825	0.787	0.584	0.734	0.595	0.933	0.914	0.732
	NBFNet	**0.845**	0.949	0.946	0.947	**0.946**	**0.897**	**0.904**	**0.889**	0.644	**0.953**	**0.967**	**0.928**
	MorsE (RotatE)	0.832	**0.957**	**0.957**	**0.959**	0.841	0.815	0.709	0.796	0.652	0.807	0.877	0.534
Rule	RuleN	0.498	0.778	0.877	0.856	0.809	0.782	0.534	0.716	0.535	0.818	0.773	0.614
	AnyBURL	0.604	0.823	0.847	0.849	0.867	0.828	0.656	0.796	**0.683**	0.835	0.798	0.652
Diff-R	DRUM	0.529	0.587	0.529	0.559	0.744	0.689	0.462	0.671	0.194	0.786	0.827	0.806
	Neural-LP	0.529	0.589	0.529	0.559	0.744	0.689	0.462	0.671	0.408	0.787	0.827	0.806
	RESHUFFLE	0.747	0.885	0.903	0.918	0.710	0.729	0.602	0.694	0.638	0.861	0.882	0.812

Table 1: Hits@10 for 50 negative samples on inductive KGC split by method type (GNN-based vs. rule-based vs. differentiable rule-based). AnyBURL and NBFNet results were obtained from Anil et al. (2024); Neural-LP, DRUM, RuleN, and GraIL results are from Teru, Denis, and Hamilton (2020); CoMPILE results are from Mai et al. (2021); and MorsE results are from Chen et al. (2022a).

	FB15k-237				WN18RR				NELL-995			
	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
RESHUFFLE^2	0.304	0.569	0.385	0.916	0.293	0.309	0.155	0.270	0.488	0.558	0.334	0.370
RESHUFFLE$_{\mathsf{nL}}$	0.744	0.890	0.903	0.917	0.698	0.685	0.618	0.682	0.627	0.738	0.886	0.815
RESHUFFLE	0.747	0.885	0.903	0.918	0.710	0.729	0.602	0.694	0.638	0.861	0.882	0.812

Table 2: Hits@10 for 50 negative samples on inductive KGC for each ablation of RESHUFFLE.

Whether it is possible to capture bounded inference for arbitrary sets of closed path rules, intersection rules, and hierarchy rules remains an open question for future work. In the basic formulation of RESHUFFLE, we cannot model inverse relations, which also means that we cannot constrain relations to be symmetric. However, in practice, for each triple (e, r, f) in the KG, we add an inverse triple (f, r^{inv}, e). Each triple thus induces two constraints $\mathbf{B_r Z_e} \preceq \mathbf{Z_f}$ and $\mathbf{B_{r^{inv}} Z_f} \preceq \mathbf{Z_e}$. The fact that r is a symmetric relation can then be straightforwardly captured by requiring $\mathbf{B_r} = \mathbf{B_{r^{inv}}}$. Disjointness constraints of the form $r_1(X, Y) \wedge r_2(X, Y) \rightarrow \bot$, indicating that r_1 and r_2 can never hold together, have also been studied in the context of region-based embeddings. RESHUFFLE does not have a mechanism to capture such constraints, but this can be addressed by adding a bias term, associating each triple (e, r, f) with constraints of the form $\mathbf{B_r Z_e} + \mathbf{c_r} \preceq \mathbf{Z_f}$ and $\mathbf{B_{r^{inv}} Z_f} + \mathbf{c_r^{\text{inv}}} \preceq \mathbf{Z_e}$.

8 Empirical Evaluation

We complement our theoretical results with an empirical evaluation, focused on showing that suitable model parameters can be effectively learned. In particular, we want to assess whether the model is simple enough to avoid overfitting, and whether it can compete with other (differentiable) rule learning methods. We focus on *inductive* KG completion, as the need to capture reasoning patterns is intuitively more important for this setting compared to the traditional (i.e. transductive) setting.[3]

Model Details We learn a soft approximation of the matrices $\mathbf{B_r}$. Specifically, we learn each row i of $\mathbf{B_r}$ by selecting the first ℓ coordinates of a vector $\mathsf{softmax}(b^r_{i,1}, ..., b^r_{i,\ell+1})$, with $b^r_{i,1}, ..., b^r_{i,\ell+1}$ learnable parameters. Note that we need $\ell + 1$ parameters to allow some rows to be all 0s, which we empirically found to be important. The number of parameters per relation is thus quadratic in ℓ. However, due to the use of the softmax operation, these representations can still be learned effectively (Lavoie et al. 2023).[4]

To initialise the entity embeddings, we set each coordinate to 0 or 1, with 50% probability. To train the model, we use the following scoring function for a given triple (e, r, f):

$$s(e, r, f) = -\|\mathsf{ReLU}(\mathbf{B_r}\, \mathbf{Z_e^{(m)}} - \mathbf{Z_f^{(m)}})\|_2$$

where m denotes the number of GNN layers. Note that $s(e, r, f) = 0$ reaches its maximal value of 0 iff $\mathbf{B_r Z_e^{(m)}} \preceq \mathbf{Z_f^{(m)}}$. For each triple (e, r, f) in the given KG $\mathcal{G}$, we add an inverse triple (f, r_{inv}, e) to $\mathcal{G}$. For each entity e, we also add the triple (e, eq, e) to $\mathcal{G}$. Following the literature (Teru, Denis, and Hamilton 2020; Zhu et al. 2021), RESHUFFLE's training process uses negative sampling under the partial completeness assumption (PCA) (Galárraga et al. 2013), i.e., for each training triple $(e, r, f) \in \mathcal{G}$, N triples (negative samples) are created by replacing e or f in (e, r, f) by randomly sampled entities $e', f' \in \mathcal{E}$. To train RESHUFFLE, we minimise the margin ranking loss, defined as follows:

$$L(e, r, f) = \sum_{i=1}^{N} \max(0, s(e'_i, r, f'_i) - s(e, r, f) + \lambda) \quad (9)$$

[3]The code for replicating our experiments is available at: https://github.com/AleksVap/RESHUFFLE.

[4]We experimented with a number of strategies for imposing sparsity, but were not able to outperform the softmax formulation.

where (e'_i, r, f'_i) is the i[th] negative sample and $\lambda > 0$ is a hyperparameter, called the margin. Intuitively, the margin ranking loss pushes scores of true triples (i.e., those within the training graph) to be larger by at least λ than the scores of triples that are likely false (i.e., negative samples).

Experimental Setup We evaluate RESHUFFLE on the inductive knowledge graph completion (KGC) benchmark that was derived by Teru, Denis, and Hamilton (2020) from FB15k-237, WN18RR, and NELL-995. For each of these KGs, four different datasets were obtained, named v1 to v4, leading to a total of twelve datasets. Each dataset consists of two disjoint graphs, a training graph $\mathcal{G}_{Train}$ and a testing graph $\mathcal{G}_{Test}$. Both of these graphs are split into a train set (80%), validation set (10%), and test set (10%), leading to a total of six graphs per dataset. Following Teru, Denis, and Hamilton (2020), we train RESHUFFLE on the train split of $\mathcal{G}_{Train}$, tune our model's hyperparameters on the validation split of $\mathcal{G}_{Train}$, and finally evaluate the best model on the test split of $\mathcal{G}_{Test}$. To account for small performance fluctuations, we repeat our experiments three times and report RESHUFFLE's average performance.[5] We select the hyperparameter configuration with the highest Hits@10 score on the validation split of $\mathcal{G}_{Train}$. Further details on hyperparameter tuning can be found in the online appendix. In accordance with Teru, Denis, and Hamilton (2020), we evaluate RESHUFFLE's test performance on 50 negatively sampled entities per triple of the test split of $\mathcal{G}_{Test}$ and report the Hits@10 scores. We list further details about the experimental setup in the online appendix.

Our GNN model acts as a kind of differentiable rule base. We therefore compare RESHUFFLE to standard approaches for differentiable rule learning: Neural-LP (Yang, Yang, and Cohen 2017) and DRUM (Sadeghian et al. 2019). We also compare our method to two classical rule learning methods: RuleN (Meilicke et al. 2018) and AnyBURL (Meilicke et al. 2019). Finally, we include a comparison with GNN-based approaches: CoMPILE (Mai et al. 2021), GraIL (Teru, Denis, and Hamilton 2020), NBFNet (Zhu et al. 2021), and MorsE (Chen et al. 2022a).

Results The results in Table 1 reveal that RESHUFFLE consistently outperforms the differentiable rule learners DRUM and Neural-LP, often by a significant margin (with WN18RR-v1 as the only exception). Compared to the traditional rule learners, RESHUFFLE performs clearly better on FB15k-237 and NELL-995 (apart from v1) but underperforms on the WN18RR benchmarks. Anil et al. (2024) found that the kinds of rules which are needed for WN18RR are much noisier compared to those than those which are needed for FB15k-237 and NELL-995. Our use of ordering constraints may be less suitable for such cases. Finally, compared to the GNN-based methods, RESHUFFLE outperforms CoMPILE and GraIL on FB15k-237 and NELL-995 v1 and v4 while again (mostly) underperforming on WN18RR. Despite the promising results compared to (differentiable) rule learners, RESHUFFLE is not competitive against state-of-the-art models such as NBFNet and MorsE. This finding is compatible with the analysis from Anil et al. (2024), which suggests that achieving state-of-the-art performance requires going beyond rule-based reasoning.

Ablation Analysis We consider two variants of our model: (i) RESHUFFLE$_{\mathsf{nL}}$, which does not add a self-loop relation to the KG (i.e. triples of the form (e, eq, e)); and (ii) RESHUFFLE^2, which allows for more general $\mathbf{B_r}$ matrices. Different from RESHUFFLE, which uses the softmax function to learn the rows of $\mathbf{B_r}$, RESHUFFLE^2 squares the $\mathbf{B_r}$ matrices component-wise, thereby allowing arbitrary positive values. For a fair comparison, we train each variant with the same hyperparameter values, experimental setup, and evaluation protocol. The results in Table 2 show that RESHUFFLE performs comparable to or better than RESHUFFLE$_{\mathsf{nL}}$ and dramatically outperforms RESHUFFLE^2 on all benchmarks. The similar performance of RESHUFFLE and RESHUFFLE$_{\mathsf{nL}}$ on most datasets suggests that the self-loop relation only matters in specific cases, which may not occur frequently in some datasets. The poor performance of RESHUFFLE^2 is as expected, since allowing arbitrary positive parameters makes overfitting the training data more likely.

9 Conclusions

The region-based view of KG embeddings makes it possible to formally analyse which inference patterns are captured by a given embedding. An important question, which was left unanswered by previous work, is whether a region-based embedding model can be found which is capable of capturing arbitrary sets of closed path rules, while still ensuring that embeddings can be learned effectively in practice. In this context, we proposed a novel approach based on ordering constraints between reshuffled entity embeddings. This model, called RESHUFFLE, was chosen because it allows us to escape the limitations of coordinate-wise approaches while otherwise remaining as simple as possible. We found that RESHUFFLE has several interesting properties. Most significantly, we showed that bounded reasoning with arbitrary sets of closed path rules can be faithfully captured. We also revealed special cases where exact reasoning is possible, which go significantly beyond what is (known to be) possible with existing region based models.

Empirically, we found our approach to be competitive with (differentiable) rule learners, while underperforming the state-of-the-art more generally. This latter finding reflects the fact that (differentiable) rule based methods are less suitable when we need to weigh different pieces of weak evidence. In such cases, when further evidence becomes available, we may want to revise earlier assumptions, which is not possible with RESHUFFLE. Developing effective models that can provably simulate non-monotonic (or probabilistic) reasoning thus remains as an important open challenge. Another interesting direction for future work would be to extend our model to relations of higher arity.

[5]Results for all seeds and the resulting standard deviations are provided in the online appendix.

Acknowledgments

This work was supported by the Vienna Science and Technology Fund [10.47379/VRG18013, 10.47379/NXT22018, 10.47379/ ICT2201], the Austrian Science Fund [10.55776/COE12], and EPSRC grant EP/W003309/1.

References

Abboud, R.; Ceylan, İ. İ.; Lukasiewicz, T.; and Salvatori, T. 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*.

Anil, A.; Gutiérrez-Basulto, V.; Ibáñez-García, Y.; and Schockaert, S. 2024. Inductive knowledge graph completion with gnns and rules: An analysis. In Calzolari, N.; Kan, M.; Hoste, V.; Lenci, A.; Sakti, S.; and Xue, N., eds., *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, 9036–9049. ELRA and ICCL.

Bach, S. H.; Broecheler, M.; Huang, B.; and Getoor, L. 2017. Hinge-loss markov random fields and probabilistic soft logic. *J. Mach. Learn. Res.* 18:109:1–109:67.

Balazevic, I.; Allen, C.; and Hospedales, T. M. 2019. 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*, 5184–5193. Association for Computational Linguistics.

Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; and Yakhnenko, O. 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*, 2787–2795.

Bourgaux, C.; Guimarães, R.; Koudijs, R.; Lacerda, V.; and Ozaki, A. 2024. Knowledge base embeddings: Semantics and theoretical properties. In Marquis, P.; Ortiz, M.; and Pagnucco, M., eds., *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR 2024, Hanoi, Vietnam. November 2-8, 2024*.

Charpenay, V., and Schockaert, S. 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*, 3289–3297. ijcai.org.

Chen, M.; Zhang, W.; Zhu, Y.; Zhou, H.; Yuan, Z.; Xu, C.; and Chen, H. 2022a. Meta-knowledge transfer for inductive knowledge graph embedding. In Amigó, E.; Castells, P.; Gonzalo, J.; Carterette, B.; Culpepper, J. S.; and Kazai, G., eds., *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, 927–937. ACM.

Chen, Y.; Mishra, P.; Franceschi, L.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2022b. ReFactor GNNs: Revisiting factorisation-based models from a message-passing perspective. In *NeurIPS*.

Chen, S.; Cai, Y.; Fang, H.; Huang, X.; and Sun, M. 2023. Differentiable neuro-symbolic reasoning on large-scale knowledge graphs. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Cheng, K.; Ahmed, N. K.; and Sun, Y. 2023. Neural compositional rule learning for knowledge graph reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Galárraga, L. A.; Teflioudi, C.; Hose, K.; and Suchanek, F. M. 2013. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In Schwabe, D.; Almeida, V. A. F.; Glaser, H.; Baeza-Yates, R.; and Moon, S. B., eds., *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, 413–422. International World Wide Web Conferences Steering Committee / ACM.

Guo, S.; Wang, Q.; Wang, L.; Wang, B.; and Guo, L. 2016. Jointly embedding knowledge graphs and logical rules. In Su, J.; Carreras, X.; and Duh, K., eds., *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 192–202. The Association for Computational Linguistics.

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

Lavoie, S.; Tsirigotis, C.; Schwarzer, M.; Vani, A.; Noukhovitch, M.; Kawaguchi, K.; and Courville, A. C. 2023. Simplicial embeddings in self-supervised learning and downstream classification. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Leemhuis, M.; Özçep, Ö. L.; and Wolter, D. 2022. Learning with cone-based geometric models and orthologics. *Ann. Math. Artif. Intell.* 90(11-12):1159–1195.

Mai, S.; Zheng, S.; Yang, Y.; and Hu, H. 2021. Communicative message passing for inductive relation reasoning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 4294–4302. AAAI Press.

Meilicke, C.; Fink, M.; Wang, Y.; Ruffinelli, D.; Gemulla, R.; and Stuckenschmidt, H. 2018. Fine-grained evaluation of rule- and embedding-based systems for knowledge graph completion. In Vrandecic, D.; Bontcheva, K.; Suárez-Figueroa, M. C.; Presutti, V.; Celino, I.; Sabou, M.; Kaffee, L.; and Simperl, E., eds., *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I*, volume 11136 of *Lecture Notes in Computer Science*, 3–20. Springer.

Meilicke, C.; Chekol, M. W.; Ruffinelli, D.; and Stuckenschmidt, H. 2019. Anytime bottom-up rule learning for knowledge graph completion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 3137–3143. ijcai.org.

Nickel, M.; Tresp, V.; and Kriegel, H. 2011. A three-way model for collective learning on multi-relational data. In Getoor, L., and Scheffer, T., eds., *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, 809–816. Omnipress.

Pavlovic, A., and Sallinger, E. 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.

Qu, M., and Tang, J. 2019. Probabilistic logic neural networks for reasoning. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 7710–7720.

Qu, M.; Chen, J.; Xhonneux, L. A. C.; Bengio, Y.; and Tang, J. 2021. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Sadeghian, A.; Armandpour, M.; Ding, P.; and Wang, D. Z. 2019. DRUM: end-to-end differentiable rule mining on knowledge graphs. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 15321–15331.

Sun, Z.; Deng, Z.; Nie, J.; and Tang, J. 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.

Tang, X.; Zhu, S.; Liang, Y.; and Zhang, M. 2024. Rule: Knowledge graph reasoning with rule embedding. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, 4316–4335. Association for Computational Linguistics.

Teru, K. K.; Denis, E. G.; and Hamilton, W. L. 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*, 9448–9457. PMLR.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In Balcan, M., and Weinberger, K. Q., eds., *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, 2071–2080. JMLR.org.

Yang, B.; Yih, W.; He, X.; Gao, J.; and Deng, L. 2015. Embedding entities and relations for learning and inference in knowledge bases. In Bengio, Y., and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yang, F.; Yang, Z.; and Cohen, W. W. 2017. Differentiable learning of logical rules for knowledge base reasoning. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2319–2328.

Zhang, Y., and Yao, Q. 2022. Knowledge graph reasoning with relational digraph. In *WWW*, 912–924. ACM.

Zhang, Z.; Wang, J.; Chen, J.; Ji, S.; and Wu, F. 2021. Cone: Cone embeddings for multi-hop reasoning over knowledge graphs. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 19172–19183.

Zhu, Z.; Zhang, Z.; Xhonneux, L. A. C.; and Tang, J. 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*, 29476–29490.

Zhu, Z.; Yuan, X.; Galkin, M.; Xhonneux, L. A. C.; Zhang, M.; Gazeau, M.; and Tang, J. 2023. A*net: A scalable path-based reasoning approach for knowledge graphs. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.