
When No Paths Lead to Rome: Benchmarking Systematic Neural Relational Reasoning

Anirban Das*
Cardiff University
dasa8@cardiff.ac.uk

Irtaza Khalid*
Cardiff University
khalidmi@cardiff.ac.uk

Rafael Peñaloza
University of Milano-Bicocca
rafael.penalozyassen@unimib.it

Steven Schockaert
Cardiff University
schockaerts1@cardiff.ac.uk

Abstract

Designing models that can learn to reason in a systematic way is an important and long-standing challenge. In recent years, a wide range of solutions have been proposed for the specific case of systematic relational reasoning, including Neuro-Symbolic approaches, variants of the Transformer architecture, and specialised Graph Neural Networks. However, existing benchmarks for systematic relational reasoning focus on an overly simplified setting, based on the assumption that reasoning can be reduced to composing relational paths. In fact, this assumption is hard-baked into the architecture of several recent models, leading to approaches that can perform well on existing benchmarks but are difficult to generalise to other settings. To support further progress in the field of systematic relational reasoning with neural networks, we introduce NoRA, a new benchmark which adds several levels of difficulty and requires models to go beyond path-based reasoning.

1 Introduction

The problem of *relational reasoning* involves predicting relationships between entities that are entailed from a given set of facts (expressing properties of different entities and how they are related). Entailment arises from a set of rules that a model must learn from examples. The central challenge lies in designing models capable of *systematic* reasoning, a concept closely linked to compositional generalization (Hupkes et al., 2020). This means that models should be able to solve test cases by applying the rules they have learned in novel ways. Recently, various neural network models have been proposed for this purpose, including neuro-symbolic approaches (Minervini et al., 2020), path-based methods (Cheng et al., 2023b), transformer variants (Bergen et al., 2021), and graph neural networks (GNNs) (Khalid and Schockaert, 2025).

Two significant problems are the lack of datasets that adequately test for systematicity, and the fact that state-of-the-art models heavily leverage the structure of existing benchmarks. CLUTRR (Sinha et al., 2019), the most popular benchmark for assessing systematicity, focuses on inferring family relationships. While all CLUTRR training examples can be solved in at most four inference steps, the test examples may require up to ten. Standard GNNs struggle with this kind of length generalization. Furthermore, the most successful neural methods exploit a specific characteristic of CLUTRR: the reasoning process reduces to composing relations along a single path connecting the target and source

*Equal contribution.

Data generation code is available at <https://github.com/axd353/WhenNoPathsLeadToRome/>
Eval code is available at <https://github.com/erg0dic/WhenNoPathsLeadToRome/>

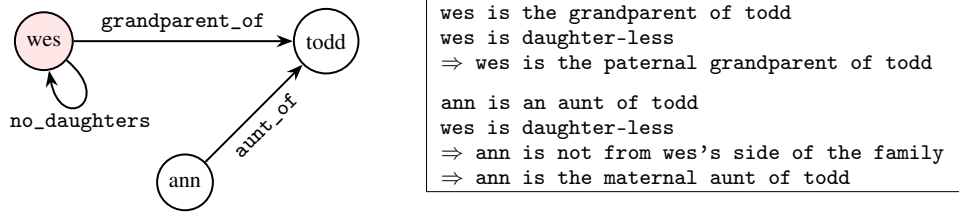


Figure 1: Example where path-based reasoning fails: to derive that ann is todd’s maternal aunt, one must consider wes, who is not on any connecting path between ann and todd.

entities, where the relational facts are viewed as a knowledge graph. For example, given the path $a \xrightarrow{\text{brother-of}} b \xrightarrow{\text{daughter-of}} c \xrightarrow{\text{brother-of}} d$, one can infer that d is the uncle of a by composing the relations *brother-of*, *daughter-of*, and *brother-of*. We refer to this style of reasoning as *path-based reasoning*. Relational reasoning often requires going beyond path-based reasoning, but this is not reflected in existing benchmarks. The only exception is STaR [Khalid and Schockaert, 2025], which focuses on temporal and spatial reasoning, and requires combining the predictions of multiple relational paths. However, the main style of reasoning that is tested by this benchmark is still path-based.

In this paper, we introduce **NoRA** (Non-Path Reasoning with Ambiguous Facts), a new benchmark which challenges state-of-the-art neural models for relational reasoning. NoRA is inspired by CLUTRR, but it intentionally breaks many of the structural assumptions in CLUTRR that state-of-the-art models are hard-coded to exploit. Like CLUTRR, the rules to be learned in NoRA are intuitive and grounded in everyday relationships—ones that humans and large language models (LLMs) naturally accept as plausible or true. However, NoRA differs from CLUTRR in three key ways.

First, NoRA is specifically designed to break the path-based inductive bias that many existing relational reasoning models rely on. To this end, NoRA considers a richer set of relationships, including more fine-grained, gender-specific family roles such as *maternal aunt of*, and everyday relations such as *is schoolmates with* and *lives in the same place as*, which often require models to go beyond path-based reasoning. Figure 1 illustrates such a case. In this example, we can infer that ann is the maternal aunt of todd, as explained in the figure, but to arrive at this conclusion, the reasoning must detour through the node wes, which is not on a path between ann and todd in the graph.

A second notable feature of NoRA is that multiple relationships may hold between a given pair of entities. These may be hierarchical (e.g. ann is both the aunt and the *maternal_aunt* of todd) or independent (e.g. a person’s brother can also be their schoolmate).

Finally, NoRA incorporates a small number of ambiguous facts in its problem instances, for instance expressing that a is the *father_of* either b or c . We argue that neural relational reasoning models should be equipped to handle such ambiguity, given its ubiquity in real-world text-based reasoning. To resolve ambiguities, a model must learn to reason with constraints: the model must evaluate multiple possibilities and then (i) eliminate any possibilities that violate constraints and (ii) determine whether a given relationship holds across all the remaining possibilities.

We make the following contributions:

- We introduce NoRA, a benchmark for systematic neural relational reasoning.
- We measure the difficulty of NoRA problem instances along a number of dimensions, corresponding to the length of inference chains, the amount of ambiguity, and the extent to which the required form of reasoning goes beyond path-based reasoning.
- We empirically show that state-of-the-art neural models for systematic reasoning struggle on NoRA, highlighting the need for new approaches.

2 Related work

The problem of learning to reason has traditionally been studied in Inductive Logic Programming [Muggleton and Raedt, 1994] (ILP). Formally, given a background theory $\mathcal{B}$ and sets of positive and negative examples, ILP considers the problem of finding a set of clauses $\mathcal{H}$ such that $\mathcal{B} \cup \mathcal{H}$ logically

entails every positive example and none of the negative examples. While important contributions in ILP continue to be made [Cropper et al., 2022], in recent years the focus has mostly shifted to neuro-symbolic methods, which try to solve the problem of learning to reasoning with a differentiable objective, for instance by simulating logic programming using tensor multiplication [Yang et al., 2017, Sadeghian et al., 2019, Dong et al., 2019], by interpreting logical connectives using fuzzy logic [Evans and Grefenstette, 2018, Sourek et al., 2018, Badreddine et al., 2022], or by using a probabilistic semantics [Manhaeve et al., 2021]. However, these approaches are mostly designed for injecting background knowledge into the training process of a neural network model, or for making one-off predictions (e.g. for knowledge graph completion), rather than for systematic reasoning.

Systematic reasoning tasks require models to learn to compose logical rules to infer conclusions. The difficulty stems from the fact that the derivations (i.e. the specific sequences of rule applications) that are needed for solving test instances differ from those in the training data, even if the training data contains sufficient information to learn all the required rules individually. Existing benchmarks that test for systematic reasoning include CLUTRR [Sinha et al., 2019], which involves predicting family relationships, GraphLog [Cohen, 2019], which involves reasoning about synthetically generated knowledge graphs, and STaR [Khalid and Schockaert, 2025], which involves qualitative temporal and spatial reasoning. CLUTRR and GraphLog can be solved by path-based reasoning, i.e. the target relationship can be inferred by selecting a single relational path between the two target nodes and composing the relations along that path. STaR requires models to compose relationships among multiple paths and then taking the intersection of the resulting predictions. Reasoning is thus more involved, although still mainly path-based. Nonetheless, these benchmarks are already challenging for most approaches. Conditional Theorem Provers (CTPs) [Minervini et al., 2020] were one of the first methods to achieve a near-perfect accuracy on CLUTRR. CTPs use a form of differentiable logic programming based on a soft unification mechanism. An important drawback of CTPs is that they are computationally expensive, which makes them impractical for many applications. Recently, a number of more efficient approaches for systematic reasoning have been proposed, such as R5 [Lu et al., 2022] and NCRL [Cheng et al., 2023a]. For benchmarks that only require path-based reasoning, these approaches can be effective, but they cannot be used in more general settings such as STaR and our proposed benchmark. Edge transformers [Bergen et al., 2021] are a modification of the transformer architecture, with a triangular attention mechanism that is designed to facilitate relational reasoning. They perform well on path-based benchmarks such as CLUTRR, albeit somewhat worse than CTPs, R5 and NCRL. In contrast to the aforementioned methods, their architecture does not constrain them to path-based reasoning. They also performed reasonably well on STaR. Finally, EpiGNNs [Khalid and Schockaert, 2025] are a type of GNN model with an inductive bias for systematic relational reasoning. Their architecture is designed to support reasoning tasks where the predictions of multiple paths need to be combined, and are thus well-suited to benchmarks such as STaR.

The problem of systematic relational learning is fundamentally different from knowledge graph (KG) completion, despite the close similarities in the format of both tasks. KG completion often requires making predictions that cannot be logically entailed, by exploiting statistical biases. Because KG completion models have to capture such biases, they typically perform poorly on systematic generalization tasks. Conversely, models that are designed for systematic reasoning tend to underperform on KG completion benchmarks; see e.g. the comparison between NBFNet [Zhu et al., 2021] and EpiGNN by [Khalid and Schockaert, 2025]). Interestingly, the fact that path-based reasoning is not always sufficient has also been highlighted in the context of KG completion [Sun et al., 2024].

3 Problem setting

Before introducing the NoRA benchmark, we introduce the problem setting and some notations.

Stories We consider the problem of reasoning about *stories*, which in this context are sets of facts. Stories may contain three types of facts. First, we have binary facts, expressing a relationship between two entities, e.g. `school_mates_with(ram, irfan)`. Second, we have unary facts, e.g. `underage(ryan)`, expressing a property of a single entity. Finally, we also have facts encoding ambiguous relationships. We use the syntax of Answer Set Programming (ASP) [Gelfond and Lifschitz

[1988]) to encode such facts² The general form of an ambiguous fact is as follows:

$$l \{r_1(x_1, y_1), \dots, r_n(x_n, y_n)\} u$$

It expresses that between l and u of the binary facts $r_1(x_1, y_1), \dots, r_n(x_n, y_n)$ are true. We will specifically use such facts to encode relationships where one of the arguments is ambiguous, e.g.:

$$1 \{r(x, y_1), r(x, y_2)\} 1 \quad (1)$$

This encodes that x is in relationship r with either y_1 or y_2 (not both). Such ambiguities often arise when reasoning about information coming from text, for instance because of ambiguous coreferences.

World rules All the stories in our dataset satisfy some regularities, which are formalized using definite rules and constraint rules. We will together refer to them as the *world rules* and again use ASP syntax. *Definite rules* allow us to infer relational facts from a given set of facts, e.g.:

$$\begin{aligned} \text{living_in_same_place}(X, Z) &:- \text{living_in_same_place}(X, Y), \text{living_in_same_place}(Y, Z). \\ \text{underage}(X) &:- \text{school_mates_with}(X, U). \end{aligned}$$

Uppercase arguments like X denote variables. The head (left side of a rule) specifies what is inferred, while the body (right side) specifies the conditions. The first rule expresses that the `living_in_same_place` relation is transitive. The second rule expresses that if somebody is school mates with somebody else, then they must be underage. *Constraint rules* specify that some sets of facts can never be true at the same time. They are encoded as rules with an empty head, e.g.:

$$:- \text{underage}(X), \text{parent_of}(X, Y).$$

This constraint expresses that underage people cannot be parents.

Answer sets The world rules allow us to reason about the facts that are specified in a given story $\mathcal{S}$. This process serves two purposes. First, the facts in the story are incomplete, in the sense that we can infer additional facts by applying the definite rules. Second, the constraints allow us to eliminate some of the ambiguity. To formally define the reasoning process, we need the concept of *answer set*³. For a story $\mathcal{S}$ without ambiguity, its answer set contains all facts inferable from $\mathcal{S}$ via definite rules. If constraints are violated by this set, $\mathcal{S}$ has no answer set; otherwise, $\mathcal{A} = \text{ans}(\mathcal{S})$ denotes $\mathcal{S}$'s answer set. Now consider a story $\mathcal{S}$ that contains the ambiguous fact $\textcircled{1}$. There are two possibilities: either $r(x, y_1)$ is true or $r(x, y_2)$ is true. Accordingly, we may consider two alternatives: the story $\mathcal{S}'$ in which the ambiguous fact $\textcircled{1}$ is replaced by $r(x, y_1)$, and the story $\mathcal{S}''$ in which the ambiguous fact is instead replaced by $r(x, y_2)$. We can repeat this process for all the ambiguous facts, leading to a set of unambiguous stories $\mathcal{S}_1, \dots, \mathcal{S}_k$. We will refer to these stories as the *refinements* of $\mathcal{S}$. Then we say that $\mathcal{A}$ is an answer set of $\mathcal{S}$ if there is an $i \in \{1, \dots, k\}$ such that $\mathcal{A} = \text{ans}(\mathcal{S}_i)$. A story with ambiguous facts may thus have 0, 1 or multiple answer sets. Let us write $\text{ref}^+(\mathcal{S})$ for the refinements of $\mathcal{S}$ which have an answer set (i.e. the different ways in which the ambiguities can be resolved without violating any constraints) and let $\text{ref}^-(\mathcal{S})$ denote the other refinements (i.e. those where the inferred facts violate some constraints).

Problem formulation The training data consists of tuples $(\mathcal{S}, x, y, \mathcal{R})$, where $\mathcal{S}$ is a story, x and y are entities that appear in $\mathcal{S}$, and $\mathcal{R}$ is the set of relationships that can be inferred to hold between x and y . Formally, let us write $\text{rels}(x, y, \mathcal{A})$ for the set of relationships that are asserted to hold between x and y in a given answer set $\mathcal{A}$:

$$\text{rels}(x, y, \mathcal{A}) = \{r \mid r(x, y) \in \mathcal{A}\}$$

then we have:

$$\mathcal{R} = \bigcap \{\text{rels}(x, y, \mathcal{A}) \mid \mathcal{A} \text{ is an answer set of } \mathcal{S}\}$$

The dataset is generated such that every story $\mathcal{S}$ has at least one answer set, i.e. there is always a way to resolve the ambiguities which is consistent with the constraints of the world. Test instances are queries of the form $(\mathcal{S}, x, y, ?)$, i.e. given a story and two designated entities x (source) and y (target), the task is to predict all relationships that can be inferred to hold between x and y . The world rules are fixed across all training and test examples. The model is thus required to induce the world rules from the training examples, and to learn to apply them in a systematic way.

²More precisely, we use the syntax of Clingo: <https://github.com/potassco>

³In general, answer sets are defined in terms of the so-called Gelfond-Lifschitz reduct [Gelfond and Lifschitz 1988]. For the simplified setting here, answer sets can be defined more straightforwardly.

Example Suppose we have a story $\mathcal{S}$ consisting of the following facts.⁴

$child_of(john,mary) \quad colleague_of(mary,bob) \quad 1\{living_in(bob,paris), living_in(bob,rome)\}1$
 $living_in(john,rome) \quad school_mate_with(john,eve) \quad 1\{child_of(eve,ann), child_of(eve,paul)\}1$

There are two ambiguous facts, which means that there may be up to four answer sets. However, from $school_mate_with(john,eve)$ we infer that $john$ is underage. We have a world rule which states that underage children live in the same place as their parents. Together with $living_in(john,rome)$ we infer $living_in(mary,rome)$. We have a rule that colleagues live in the same place, allowing us to infer $living_in(bob,rome)$. The option $living_in(bob,paris)$ is thus not consistent with the available facts (we have a constraint stating that people cannot live in two different places). The other ambiguity cannot be resolved, so the story has two answer sets. For the query $(\mathcal{S}, mary, rome, ?)$ the answer is thus $\mathcal{R} = \{living_in\}$, as $living_in(mary,rome)$ is included in both answer sets. For the query $(\mathcal{S}, eve, ann, ?)$ the answer is $\mathcal{R} = \emptyset$, as $child_of(eve,ann)$ is only included in one of the answer sets.

4 Dataset construction

We now present the details of our benchmark and introduce a number of metrics for measuring different aspects of problem difficulty. These difficulty measures are then used for creating systematic test splits, which will allow us to evaluate different aspects of compositional generalization.

4.1 Data generation process

We generate a story by randomly generating story facts. We use Clingo 5.7.1 (Gebser et al., 2011) to obtain the answer sets. An *entailed atom* is an atom that appears in all the answer sets of the story but is not explicitly provided as a story fact. The entailed atoms are used to construct the queries in our benchmark. We only retain stories that have at least one answer set and at least one entailed atom. We generate many stories, where each story includes a different set of story facts, while the world rules remain constant across all generated stories for a dataset. Details regarding the exact world rules, types of ambiguous facts considered, and the sampling process are in Appendix D.

4.2 Measuring problem difficulty

We propose a number of metrics for measuring the difficulty of a given problem instance. These metrics serve two purposes. First, since we want to test for systematicity, we will consider test instances that are strictly harder than the training instances. To solve such test instances, models need to learn to compose the knowledge they have learned in novel ways (rather than learning shortcuts or memorizing computation graphs). Second, the proposed difficulty metrics will allow us to analyze model performance in a more fine-grained way.

Reasoning depth A standard notion of difficulty is the number of inference steps that are needed to infer the answer (i.e. the number of rule applications). Let $\mathcal{S}$ be a given story, and let $\mathcal{S}_1, \dots, \mathcal{S}_k$ be the refinements of $\mathcal{S}$ that are consistent with the constraints. The answer sets of $\mathcal{S}$ are then given by $\mathcal{A}_1, \dots, \mathcal{A}_k$ with $\mathcal{A}_i = ans(\mathcal{S}_i)$. Let $r(a, b)$ be a fact that is included in $\mathcal{A}_i$. We define the reasoning depth of $r(a, b)$ in $\mathcal{S}_i$, written $depth(r(a, b), \mathcal{S}_i)$, as the minimum number of inference steps that are needed to infer $r(a, b)$ from $\mathcal{S}_i$. For instance, if $r(a, b)$ is included in $\mathcal{S}_i$, we have $depth(r(a, b), \mathcal{S}_i) = 0$. Similarly, if $\mathcal{S}_i$ is a refinement that violates the constraints, we write $depth(\perp, \mathcal{S}_i)$ for the minimum number of inference steps that are needed to establish that the constraints are violated. The *maximum reasoning depth* of a problem instance $(\mathcal{S}, a, b, \mathcal{R})$ is then computed as follows:

$$max_depth(\mathcal{S}, a, b, \mathcal{R}) = \max \left(\{depth(r(a, b), \mathcal{S}_i) \mid \mathcal{S}_i \in ref^+(\mathcal{S}), r \in \mathcal{R}\} \right. \\ \left. \cup \{depth(\perp, \mathcal{S}_i) \mid \mathcal{S}_i \in ref^-(\mathcal{S})\} \right)$$

The reasoning depth is determined by the hardest relation in $\mathcal{R}$ and the hardest answer set.

⁴More elaborate examples can be found in the Appendix C.

Reasoning width Intuitively, the more ambiguity in a given problem instance, the harder it is to solve, all things being equal. We can straightforwardly measure the amount of ambiguity by computing the number of possible refinements of a story $\mathcal{S}$. If each ambiguous fact introduces two possibilities, then the number of possible refinements is 2^N , with N the number of ambiguous facts⁵. However, some ambiguous fact may not play any role in the derivation of the query, so simply counting the number of refinements may be misleading. As an alternative, we therefore focus on counting the number of unique derivations. In particular, we define the reasoning width of a fact $r(a, b)$ w.r.t. a story $\mathcal{S}$ as:

$$\text{width}(r(a, b), \mathcal{S}) = |\{\text{proof}(r(a, b), \mathcal{S}_i) \mid \mathcal{S}_i \in \text{ref}^+(\mathcal{S})\}| + |\{\text{proof}(\perp, \mathcal{S}_i) \mid \mathcal{S}_i \in \text{ref}^-(\mathcal{S})\}|$$

where we write $\text{proof}(r(a, b), \mathcal{S}_i)$ for the derivation which proves that $r(a, b)$ can be derived from $\mathcal{S}_i$. If there are multiple proofs, we fix $\text{proof}(r(a, b), \mathcal{S}_i)$ to be the shortest one, with ties broken arbitrarily. Similarly, $\text{proof}(\perp, \mathcal{S}_i)$ denotes a minimal proof that $\mathcal{S}_i$ violates the constraints. In other words, the width of $r(a, b)$ is the sum of the number of distinct derivations of $r(a, b)$, across all the answer sets of $\mathcal{S}$, and the number of distinct derivations of constraint violation, across all refinements of $\mathcal{S}$ without an answer set. The maximal width of a problem instance $(\mathcal{S}, a, b, \mathcal{R})$ is then computed as the reasoning width for the hardest relation in $\mathcal{R}$:

$$\text{max-width}(\mathcal{S}, a, b, \mathcal{R}) = \max\{\text{width}(r(a, b), \mathcal{S}) \mid r \in \mathcal{R}\}$$

Non-path reasoning In the case of CLUTRR, all the required rules are of the following form

$$r(X, Z) : -r_1(X, Y), r_2(Y, Z) \quad (2)$$

If all rules are like this, then the problem of inferring a relational fact $s(a, b)$ boils down to (i) finding an informative path connecting a and b (where we view the facts as the edges of a knowledge graph), and (ii) repeatedly applying rules of the form (2) to replace two adjacent edges by a single edge (representing the composition of the two given relations), until we end up with a single edge connecting a and b ⁶. Many approaches for systematic relational reasoning are closely aligned with this idea. As such, problem instances that require going beyond this kind of path-based reasoning can be expected to present difficulties for many models. We introduce two metrics to measure the extent to which a problem instance requires going beyond path-based reasoning.

The first metric, **backtrack load (BL)**, is based on the observation that for path-based derivations, the number of inference steps is always one less than the number of entities involved in the derivation. In contrast, for more complex derivations, we often see a higher number of inference steps, relative to the number of entities. We thus define $BL(\tau)$ for a derivation τ as the ratio of the number of inference steps to the number of entities involved. The maximum backtrack load of a problem instance is then:

$$\text{max-BL}(\mathcal{S}, a, b, \mathcal{R}) = \max\{BL(\text{proof}(r(a, b), \mathcal{S}_i)) \mid \mathcal{S}_i \in \text{ref}^+(\mathcal{S}), r \in \mathcal{R}\}$$

The second metric is called **off-path edge count (OPEC)**. For a given derivation τ of a fact $r(a, b)$, we define $OPEC(\tau)$ as the number of edges that appear in τ which are not on any direct path between a and b (if we view relational facts as the edges of a knowledge graph). We then define the maximal OPEC of a problem instance as:

$$\text{max-OPEC}(\mathcal{S}, a, b, \mathcal{R}) = \max\{OPEC(\text{proof}(r(a, b), \mathcal{S}_i)) \mid \mathcal{S}_i \in \text{ref}^+(\mathcal{S}), r \in \mathcal{R}\}$$

We drop the prefix max- and refer to these objects as BL and OPEC. Figure 2 illustrates how OPEC measures the extent of non-path reasoning.

These two metrics are complementary. BL captures whether the reasoning process needs to go back-and-forth along a given relational path. This back-and-forth reasoning is often required for the problems in our benchmark, even when all the edges involved are on a single path between the two query entities. We expect this to be challenging for many approaches, especially methods such as NCRL and R5 which by design only make a single pass over a given path. OPEC captures whether any off-path reasoning is required. Path-based models typically ignore any edges that are not on a direct path between the two query entities. For more discussion see Appendix 1.

4.3 Training distribution and held-out test sets

⁵This follows because the alternatives occurring in different ambiguous facts never overlap in our dataset.

⁶See Khalid and Schockaert [2025] for a formal proof of this claim.

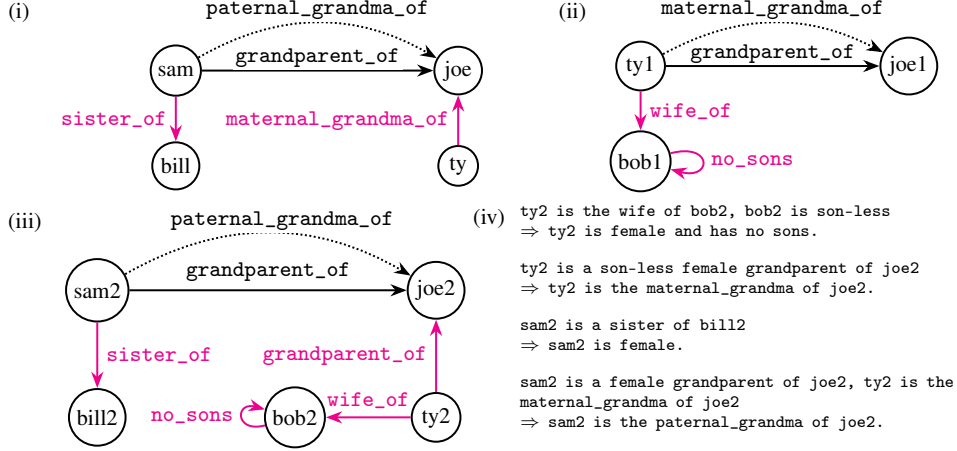


Figure 2: Source entities are *sam*, *ty1*, and *sam2*, while target entities are *joe*, *joe1*, and *joe2* for the queries accompanying stories (i), (ii) and (iii), respectively. Solid edges represent the relationships explicitly in the story. Dashed edges are entailed relationships between source–target pairs. Pink edges indicate edges that do not lie on any path between the source and target. Panel (iv) illustrates a derivation of the entailed fact in story (iii). It uses all four off-path edges, hence the query from story (iii) has an OPEC value of 4. The queries in stories (i) and (ii) each have an OPEC value of 2.

The training set for NoRA contains examples whose difficulty, according to the four proposed metrics, is controlled: reasoning depth ≤ 6 , reasoning width ≤ 5 , BL ≤ 1.5 and OPEC ≤ 2 . The marginal distribution of these four metrics within the training dataset covers a variety of examples (Appendix 9), which is essential for enabling models to generalize systematically. We have also created a separate training set which is free of ambiguity, i.e. where all examples have reasoning width 1. To rigorously test generalization, we define several held-out evaluation subsets, each focused on specific types of reasoning that go beyond what the model encounters during training. We have four such out-of-distribution test sets involving ambiguities and three which do not. Each of these out-of-distribution test sets extends the difficulty level of the problem instances according to one of the considered difficulty metrics. Finally, we also created in-distribution test sets, containing unseen problem instances with similar characteristics as those from the training set. An overview of the different datasets is shown in Table 1.

Table 1: Overview of the dataset splits. Values that require the model to generalize from the training distributions are highlighted in red.

Name	Depth	Width	BL	OPEC
Train-a	≤ 6	≤ 5	≤ 1.5	≤ 2
Train-na	≤ 6	1	≤ 1.5	≤ 2
Test-D	> 6	≤ 5	≤ 1.5	≤ 2
Test-W	≤ 6	> 5	≤ 1.5	≤ 2
Test-BL	≤ 6	≤ 5	> 1.5	-
Test-OPEC	-	-	-	> 3
Test-In-dist	≤ 6	≤ 5	≤ 1.5	≤ 2
Test-D-na	> 6	1	≤ 1.5	≤ 2
Test-BL-na	≤ 6	1	> 1.5	-
Test-OPEC-na	-	1	-	> 3
Test-In-dist-na	≤ 6	1	≤ 1.5	≤ 2

5 Experiments

We evaluate a number of state-of-the-art models on NoRA. Pure path-based methods, such as NCRL and R5, are limited to path-based inference by design, and are thus not suitable. CTPs are too inefficient to handle the large number of rules that needs to be learned for NoRA, and they cannot model constraints. We therefore focus our analysis on the following methods. **Edge Transformers** (ETs) (Bergen et al. 2021) are more versatile than other methods for systematic reasoning, and thus a natural candidate for the more challenging setting presented by NoRA. However, they cannot naturally model multiple relationships between the same entities (i.e. the edge index cannot have degeneracies). We therefore consider two versions of ETs: a vanilla ET, where a single relationship is chosen for each entity pair, arbitrarily, and others are simply ignored (single-edge) and a modified ET

Table 2: Results of state-of-the-art models for systematic reasoning on the NoRA test sets.

		Trained with ambiguity					Trained without ambiguity			
		In-dist	D	W	BL	OPEC	In-dist-na	D-na	BL-na	OPEC-na
Exact-match Accuracy	ET (single-edge)	0.885	0.741	0.703	0.245	0.060	0.800	0.822	0.104	0.110
	ET (multi-edge)	0.900	0.493	0.790	0.785	0.037	0.800	0.494	0.056	0.077
	RAT (single-edge)	0.721	0.494	0.615	0.234	0.042	0.800	0.493	0.092	0.094
	RAT (multi-edge)	0.900	0.676	0.668	0.540	0.028	0.827	0.768	0.023	0.017
	EpiGNN-min (margin)	0.334	0.491	0.176	0.000	0.000	0.208	0.485	0.000	0.000
	EpiGNN-min (BCE)	0.451	0.665	0.456	0.154	0.005	0.475	0.488	0.008	0.025
	EpiGNN-mu1 (BCE)	0.520	0.604	0.491	0.156	0.009	0.539	0.716	0.027	0.045
	NBFNet (margin)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	NBFNet (BCE)	0.576	0.531	0.460	0.153	0.009	0.679	0.764	0.012	0.043
	R-GCN	0.347	0.672	0.283	0.051	0.032	0.579	0.740	0.018	0.012

in which the edge embeddings are averaged if there are multiple relationships (multi-edge). We also evaluate transformers with **relation-aware self attention** (RAT) [Shaw et al., 2018], as a precursor to ETs. Next, we evaluate **EpiGNNs** [Khalid and Schockaert, 2025], which are the state-of-the-art on STaR (the only existing benchmark that goes beyond path-based systematic reasoning). We consider two variants: one with the original margin loss and one with a binary cross-entropy loss, with the latter intuitively being more suitable for the multi-label setting. We consider both minimum and multiplication for aggregation. Finally, we evaluate **NBFNet** [Zhu et al., 2021] and **R-GCNs** [Schlichtkrull et al., 2018] as representative GNN models. To evaluate these models, we encode ambiguities in the graph representation of stories using special edges (Appendix D).

Main results The results are shown in Table 2 in terms of exact-match accuracy (i.e. we measure if the model’s prediction of the relation set $\mathcal{R}$ exactly matches the ground truth). Models trained on *Train-a* are evaluated on the test sets with ambiguity, while models trained on *Train-na* are evaluated on the remaining test sets. ETs emerge as the best-performing model. All models perform poorly on OPEC, BL-na and OPEC-na. Surprisingly, for most models, performance on test-W is reasonable. Furthermore, all models perform better on BL than on BL-na, despite the fact that BL was assumed to be harder. Further analysis has shown that models are exploiting shortcuts to solve the majority of ambiguous problems (see Appendix K). The GNN methods all perform poorly on the BL and OPEC test sets, which can be explained by their strong alignment with path-based reasoning. In fact, the GNN models are even performing poorly on the in-distribution test sets, for the same reason. Among the GNN models, EpiGNNs with BCE loss and multiplication-based aggregation perform better. The results also confirm that the margin-based loss is unsuitable for the multi-label setting.

Analysis of ET performance Figure 3 breaks down the performance of the Edge Transformer on test-D, test-W and test-OPEC. Surprisingly, the performance decline is minimal along the considered difficulty axes. For instance, in the case of Test-D, the results for reasoning depth 12 are almost as good as those for reasoning depth 7, for the vanilla model. Similarly, apart from the dip at depth 7, the multi-edge model performs similarly between depths 8 to 12. However, these results have to be interpreted with caution. Recall that the test problems were obtained by random sampling. Obtaining hard instances in this way is difficult, meaning that we cannot easily test how the model would perform when the reasoning depth is higher than 12 or OPEC is higher than 4, for instance. This is something that we have addressed by introducing a variant of our benchmark, called NoRA v1.1, as explained below. Another consequence of the fact that randomly sampled problem instances are rarely hard relates to the correlations between the difficulty metrics. For instance, a problem with high reasoning depth will typically have low OPEC, and a problem with high OPEC will typically have low reasoning depth. Problems with high reasoning depth may thus be solved well because they are easier in other respects, rather than because the generalization abilities of the model. We analyze this in Figure 4a, where we show the performance for different reasoning depths, *while controlling for both BL and reasoning width*. In this case, we can see a dramatic decline in performance when going from reasoning depth 4 (where the multi-edge ET achieves accuracies above 0.8) to reasoning depth 6 (where the performance varies from around 0.2 to 0.6). Interestingly, in this analysis, the multi-edge variant also clearly outperforms the single-edge variant. This reflects the need for more informationally complete input representations for problems with higher BL.

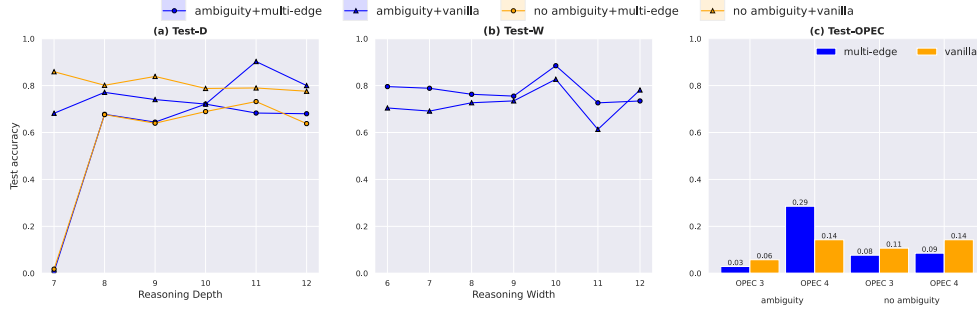


Figure 3: Analysis of the performance of ETs on various splits of the dataset.

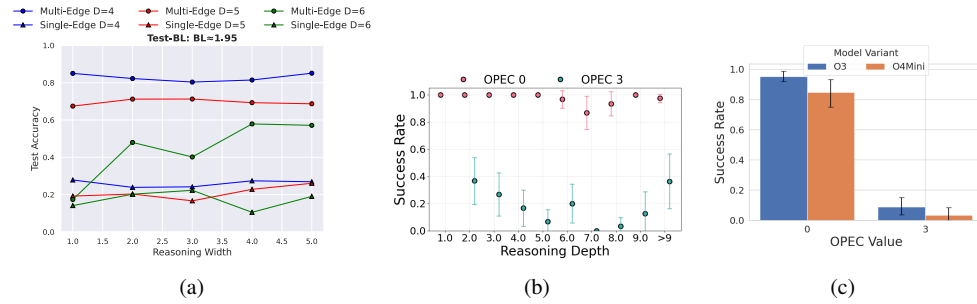


Figure 4: (a) Breakdown of the performance of edge transformers on Test-D; (b) analysis of o3 on non-ambiguous stories; (c) a comparison between o3 and o4-mini on non-ambiguous stories.

Evaluating Large Reasoning Models NoRA was designed to test the compositional generalization abilities of neural systematic reasoning models. The defined NoRA world rules are realistic, as most humans and large language models would deem them true or likely (see Appendix E). This is a desirable property for evaluating the systematic generalization and rule learning capabilities of Large Reasoning Models (LRMs) [Zhu et al., 2023]. We also evaluated the LRMs o3 and o4-mini on a subset of NoRA problems, *when explicitly given the entire set of world rules* (only in the LRM experiments are the rules explicitly provided; in all other settings, they must be induced by the model). Being able to apply the correct rules is clearly a prerequisite for solving the considered learning tasks. For this experiment (details in Appendix G), we only consider problem instances without ambiguity, as we want to focus on the extent to which these models can deal with off-path reasoning, and we only consider problem instances where there is a single best label. We provide the models with two in-context demonstrations. Success is measured by exact match with the ground truth label. The results for o3 are shown in Figure 4b. While the model achieves near-perfect accuracy for problem instances with OPEC 0, the performance drops dramatically for OPEC 3, where none of the problem instances of reasoning depth 7 were answered correctly. Even when the world rules are explicitly provided—and are rules the LRM is already familiar with through pretraining—the model fails to apply them correctly to problem instances, highlighting the inherent difficulty of off-path reasoning tasks. Surprisingly, for higher inference depths, the performance is slightly better. This is due to the presence of instances where the LRM can apply shortcuts (Appendix H). As shown in Figure 4c, the performance of o4-mini is slightly worse than that of o3. This non-path-based reasoning analysis aligns with findings from Dziri et al. [2023] on pure reasoning tasks. As an auxiliary task, we tested the model’s ability to recover the necessary world rules for solving the task (Appendix G).

Additional datasets: NoRA v1.1 and HetioNet To further support future work on neural relational reasoning, we introduce two additional datasets. First, we introduce a variant of NoRA, called NoRA v1.1, where problem instances are sampled in a more systematic way, using the recursive subgraph expansion technique [Khalid and Schockaert 2025]. This has two consequences. First, it means that we can easily create problem instances with larger OPEC, reasoning depth, and BL values. As a result, we can include examples with higher structural complexity in the training set (e.g. allowing OPEC

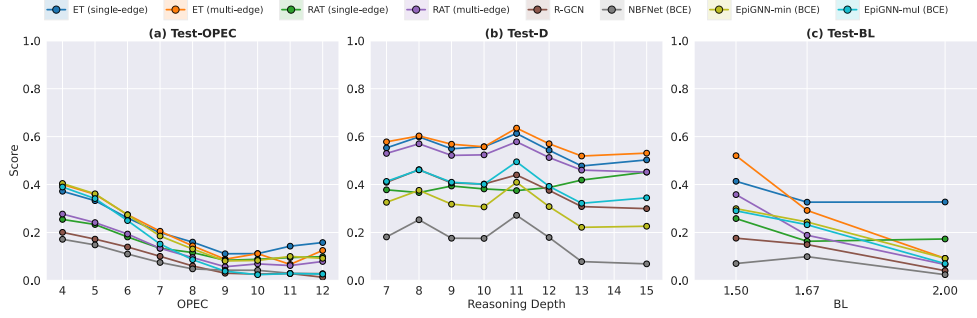


Figure 5: Results for the expanded version of NoRA (v1.1) that uses recursive subgraph expansion to generate harder splits along the axes: (a) OPEC, (b) Reasoning Depth (c) BL.

values up to 3), and include much harder problem instances in the test sets. Second, by generating the problem instances in this way, we can guarantee that every test example can be obtained by a *stitching together* process of multiple training examples. As a result, we are guaranteed that a model which achieves compositional generalization can solve every test instance. To illustrate this stitching together process, consider Figure 2 panels (i) and (ii) depict training instances with OPEC 2. These are combined by (a) deleting the fact `maternal_grandma_of(ty, joe)` from story (i), (b) renaming entities in story (ii) to align with those in story (i) (`joe1` → `joe` etc), and (c) adding the story facts from (ii) to (i). Finally renaming all entities, we obtain story (iii), which is a problem instance with OPEC 4 and is included in the test set. For NoRA v1.1 we did not include any problems with ambiguity. Figure 5 shows the performance of models on NoRA v1.1. The main conclusions from Table 2 remain valid. This demonstrates that the inability of models to handle off-path reasoning remains robust to variations in how the problem instances are generated.

We also introduce another dataset, called **HetioNet**, which was inspired by Himmelstein et al. [2023]. This dataset is based on a completely different set of world rules, unrelated to family relationships. Here, entities correspond to *diseases*, *genes*, and *drugs*, and relations capture biological phenomena. Moreover, the kinds of regularities that models are expected to learn are rather different. For instance, while families are organized into hierarchical structures, no such structure exists in the case of HetioNet. A detailed analysis of HetioNet is provided in Appendix B.2. It shows that, even after this shift in relational regularity, most models continue to struggle on tasks that require off-path reasoning. Surprisingly, however, the EpiGNN outperforms ETs on the test-OPEC split of HetioNet. Further work is needed to better understand the kinds of regularities that different models are able to capture.

6 Conclusions

We have introduced a new benchmark for systematic relational reasoning, called NoRA. It has three core features which makes it more challenging than existing benchmarks: the need for off-path reasoning, the presence of ambiguities, and the fact that entities can be simultaneously related in different ways. We found all methods to struggle significantly with off-path reasoning, suggesting that fundamentally different architectures may be needed to push forward the state-of-the-art in neural relational reasoning. Interestingly, Large Reasoning Models such as o3 were not able to solve problem instances that require off-path reasoning either, even when explicitly given all the required rules. Surprisingly, the presence of ambiguity did not pose any particular challenges for the tested models. However, further analysis revealed this to be due to the presence of shortcuts, allowing models to solve these problem instances without actually needing to reason about ambiguity. This highlights the challenge in generating hard problem instances. Finally, to test the robustness of our findings, we introduced two additional datasets: NoRA v1.1 and HetioNet.

Limitations The ability to measure the difficulty of problem instances is important for testing models in a systematic way. However, metrics such as *reasoning depth* and BL are sensitive to the way in which the knowledge base has been encoded. In the experiments with o3 we saw examples where the “true” reasoning depth was lower than that measured by the metric. All ambiguities are not equally challenging, which is something that *reasoning width* only partially captures.

Acknowledgements This work was supported by EPSRC grant EP/W003309/1.

References

- Samy Badreddine, Artur S. d’Avila Garcez, Luciano Serafini, and Michael Spranger. Logic tensor networks. *Artif. Intell.*, 303:103649, 2022. doi: 10.1016/J.ARTINT.2021.103649. URL <https://doi.org/10.1016/j.artint.2021.103649>
- Leon Bergen, Timothy J. O’Donnell, and Dzmitry Bahdanau. Systematic generalization with edge transformers. In Marc’Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 1390–1402, 2021. URL <https://proceedings.neurips.cc/paper/2021/hash/0a4dc6dae338c9cb08947c07581f77a2-Abstract.html>
- Kewei Cheng, Nesreen K. Ahmed, and Yizhou Sun. 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, 2023a. URL <https://openreview.net/pdf?id=F8VKQyDgRVj>
- Kewei Cheng, Nesreen K. Ahmed, and Yizhou Sun. 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, 2023b. URL <https://openreview.net/forum?id=F8VKQyDgRVj>
- William W. Cohen. Graphlog: A benchmark for logical learning on graphs. In *Proceedings of the 2019 International Symposium on Inductive Logic Programming*, 2019.
- Andrew Cropper, Sebastijan Dumancic, Richard Evans, and Stephen H. Muggleton. Inductive logic programming at 30. *Mach. Learn.*, 111(1):147–172, 2022. doi: 10.1007/S10994-021-06089-1. URL <https://doi.org/10.1007/s10994-021-06089-1>
- Anirban Das, Manfred Denker, Anna Levina, and Lucia Tabacu. A monte carlo algorithm for multiple stochastic integrals of stable processes. *Stochastics & Dynamics*, 22(8), 2022.
- Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. Neural logic machines. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL <https://openreview.net/forum?id=B1xY-hRctX>
- Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36:70293–70332, 2023.
- Richard Evans and Edward Grefenstette. Learning explanatory rules from noisy data. *J. Artif. Intell. Res.*, 61:1–64, 2018. doi: 10.1613/JAIR.5714. URL <https://doi.org/10.1613/jair.5714>
- Martin Gebser, Benjamin Kaufmann, Roland Kaminski, Max Ostrowski, Torsten Schaub, and Marius Schneider. Potassco: The potsdam answer set solving collection. *Ai Communications*, 24(2): 107–124, 2011.
- Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In Robert A. Kowalski and Kenneth A. Bowen, editors, *Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, USA, August 15-19, 1988 (2 Volumes)*, pages 1070–1080. MIT Press, 1988.
- Daniel S Himmelstein, Michael Zietz, Vincent Rubinetti, Kyle Kloster, Benjamin J Heil, Faisal Alquaddoomi, Dongbo Hu, David N Nicholson, Yun Hao, Blair D Sullivan, et al. Hetnet connectivity search provides rapid insights into how biomedical entities are related. *GigaScience*, 12: giad047, 2023.

- Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.
- Muhammad Khalid and Steven Schockaert. Systematic relational reasoning with epistemic graph neural networks. *The Thirteenth International Conference on Learning Representations*, 2025.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2017.
- Anna Levina and Viola Priesemann. Subsampling scaling. *Nature communications*, 8(1):15140, 2017.
- Vladimir Lifschitz. Twelve definitions of a stable model. In *International Conference on Logic Programming*, pages 37–51. Springer, 2008.
- Shengyao Lu, Bang Liu, Keith G. Mills, Shangling Jui, and Di Niu. R5: rule discovery with reinforced and recurrent relational reasoning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL <https://openreview.net/forum?id=2eXhNpHeW6E>.
- Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Neural probabilistic logic programming in deepproblog. *Artif. Intell.*, 298:103504, 2021. doi: 10.1016/J.ARTINT.2021.103504. URL <https://doi.org/10.1016/j.artint.2021.103504>.
- Pasquale Minervini, Sebastian Riedel, Pontus Stenetorp, Edward Grefenstette, and Tim Rocktäschel. Learning reasoning strategies in end-to-end differentiable proving. 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 6938–6949. PMLR, 2020. URL <http://proceedings.mlr.press/v119/minervini20a.html>.
- Stephen H. Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *J. Log. Program.*, 19/20:629–679, 1994. doi: 10.1016/0743-1066(94)90035-3. URL [https://doi.org/10.1016/0743-1066\(94\)90035-3](https://doi.org/10.1016/0743-1066(94)90035-3).
- Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. DRUM: end-to-end differentiable rule mining on knowledge graphs. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *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 15321–15331, 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/0c72cb7ee1512f800abe27823a792d03-Abstract.html>.
- Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer, 2018. doi: 10.1007/978-3-319-93417-4_38. URL https://doi.org/10.1007/978-3-319-93417-4_38.
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 464–468. Association for Computational Linguistics, 2018. doi: 10.18653/V1/N18-2074. URL <https://doi.org/10.18653/v1/n18-2074>.
- Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. Clutrr: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of EMNLP-IJCNLP*, pages 4505–4514. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1458. URL <https://doi.org/10.18653/v1/D19-1458>.
- Gustav Sourek, Vojtech Aschenbrenner, Filip Zelezný, Steven Schockaert, and Ondrej Kuzelka. Lifted relational neural networks: Efficient learning of latent relational structures. *J. Artif. Intell. Res.*, 62: 69–100, 2018. doi: 10.1613/JAIR.1.11203. URL <https://doi.org/10.1613/jair.1.11203>.

- Wangtao Sun, Shizhu He, Jun Zhao, and Kang Liu. From chain to tree: Refining chain-like rules into tree-like rules on knowledge graphs. *CoRR*, abs/2403.05130, 2024. doi: 10.48550/ARXIV.2403.05130. URL <https://doi.org/10.48550/arXiv.2403.05130>.
- Fan Yang, Zhilin Yang, and William W. Cohen. Differentiable learning of logical rules for knowledge base reasoning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2319–2328, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/0e55666a4ad822e0e34299df3591d979-Abstract.html>.
- Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural Bellman-Ford networks: A general graph neural network framework for link prediction. *Advances in neural information processing systems*, 34:29476–29490, 2021.
- Zhaocheng Zhu, Yuan Xue, Xinyun Chen, Denny Zhou, Jian Tang, Dale Schuurmans, and Hanjun Dai. Large language models can learn rules. *arXiv preprint arXiv:2310.07064*, 2023.