Journal Title
XX(X):2–27

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

NSORN: Designing a Benchmark Dataset for Neurosymbolic Ontology Reasoning with Noise

Julie Loesch[1], Gunjan Singh[2], Raghava Mutharaju[3] and Remzi Celebi[1]

Abstract

In the field of neurosymbolic computing, systematic evaluation of ontology reasoning systems under noisy conditions remains underexplored. In particular, there is a need for benchmark frameworks that enable reproducible assessment of how neurosymbolic reasoners behave when ontological data are corrupted. Thus, this work aims to develop a mechanism for introducing noise into an ontology, particularly focusing on the ABox, and evaluate the performance of existing neurosymbolic reasoners on commonly used ontologies under varying levels of noise. We developed *NSORN* (Neurosymbolic Ontology Reasoning with Noise), a first-step benchmark framework that consists of three techniques to introduce noise into ontologies: logical, statistical, and random noise. Logical noise uses logical violations of disjoint axioms and domain/range constraints. While random noise corrupts existing triples by replacing either the subject or object of a triple with a random entity, statistical noise is introduced using Graph Neural Networks to add noisy facts with low-probability scores. We evaluated the performance of existing neurosymbolic reasoners by introducing noise to *Pizza*, *Family* and *OWL2Bench* ontologies under these noise types with various levels. The resulting benchmarks were tested on two state-of-the-art neurosymbolic reasoners, *Box2EL* and *OWL2Vec**, as well as a purely neural method, *R-GCN*. We focus on reasoning tasks, such as class membership and object property assertions, to test how these reasoners handle noise. In our experimental setup, R-GCN consistently outperforms Box2EL and OWL2Vec* in robustness to noisy ontological data—even under the most destructive form, logical noise—maintaining stable membership and object property assertion performance while embedding-based models degrade sharply.

Keywords

Neurosymbolic Artificial Intelligence, Benchmark, Noise Injection, Ontology Reasoning

Introduction

Neurosymbolic computing has emerged as a prominent area of Artificial Intelligence in recent years, combining the robust learning capabilities of neural networks with the reasoning capabilities and interpretability of symbolic systems (Garcez et al., 2015; Yu et al., 2023). Symbolic reasoners rely on formal logic, rules, and knowledge bases, such as ontologies to make inferences. An ontology is a formal and explicit specification of a shared conceptualization of a domain. It describes the concepts, categories, relationships, and rules that structure knowledge within that domain, facilitating common understanding and interoperability among systems and people (Gruber, 1993). Ontologies distinguish between the Terminological Box (TBox), which defines classes, along with their relationships and restrictions, and the Assertional Box (ABox), which contains assertions about individuals such as property assertions or class memberships. Ontologies are often reliable and interpretable, offering traceable mechanisms for their inferences. However, they are sensitive to noise and struggle to handle incomplete or ambiguous data. Symbolic reasoners could fail to perform when faced with missing knowledge or errors in their knowledge base. Moreover, their reliance on a large number of predefined rules and axioms limits their scalability (Sheth et al., 2023; Makni et al., 2021). In contrast, neural reasoners leverage deep learning models, which can generalize from large volumes of data, are robust to noise. However, their primary limitation lies in their lack of interpretability (Ebrahimi et al., 2021a) and handling tasks that require explicit logic or when dealing with rare or unseen examples. Neurosymbolic reasoners can address these shortcomings inherent in each paradigm (Yu et al., 2023). By integrating symbolic reasoning with neural systems, these reasoners achieve a trade-off between interpretable logical reasoning and the scalable, data-driven capabilities of neural networks (Sarker et al., 2021; Makni et al., 2021). Despite these advantages, neurosymbolic systems face unique challenges, particularly in the incorporation of domain ontologies while ensuring resilience against the noise and uncertainty that characterize real-world data.

Noise in ontologies encompasses various forms of disturbance that can affect their integrity, coherence, and interpretability. Makni et. al. (Anke et al., 2019), presented a Semantic Web noise taxonomy, which distinguishes between two

[1]Department of Advanced Computing Sciences, Maastricht University, Netherlands
[2]FIZ Karlsruhe, Karlsruhe, Germany
[3]Mehta Family School for Data Science and Artificial Intelligence, IIT-Palakkad, India

Corresponding author:
Julie Loesch, Department of Advanced Computing Sciences, Maastricht University, Netherlands
Email: julie.loesch@maastrichtuniversity.nl

Prepared using sagej.cls

main categories of noise: TBox noise and ABox noise (i.e., propagable and non-propagable). TBox noise is the type of noise that resides within the ontology, such as in the class hierarchy, or domain and range properties. This type of noise will affect the inference over the entire dataset. While ABox noise is about corrupting an existing triple in an ontology by changing one of the triples' resources. This either changes the inference graph (i.e., propagable noise) or does not have any impact on the inference graph (i.e., non-propagable noise).

This work aims to develop *NSORN* (Neurosymbolic Ontology Reasoning with Noise), a framework designed to introduce noise into ontologies and create challenging benchmark datasets to test the effectiveness of neurosymbolic reasoners in handling noise. While numerous benchmark datasets exist for various AI tasks, such as image classification (i.e., MNIST (Deng, 2012), CIFAR-10 and CIFAR-100[*]), natural language processing (i.e., GLUE (Wang et al., 2019)), and reinforcement learning (i.e., OpenAI Gym (Brockman et al., 2016)), there is limited work on benchmark frameworks specifically tailored to evaluating neurosymbolic reasoning under noisy conditions. Such benchmark frameworks can support more systematic and comparable evaluation in this area (Raji et al., 2021). Existing neurosymbolic benchmark datasets are predominantly designed to assess the performance of symbolic reasoners (Singh, 2023a). Furthermore, most reasoning systems are evaluated using various publicly available ontologies (Singh et al., 2021, 2022; Banerjee et al., 2023), which do not address the unique challenges of neurosymbolic integration. Similarly to previous work (Liu et al., 2024), we developed three techniques to introduce noise into ontologies: logical, statistical, and random noise.

Random noise serves as a baseline, representing data-agnostic, unpredictable errors that may arise accidentally in real-world ontologies. We simulate this by corrupting existing triples—replacing either the subject or the object with a random entity. This allows us to probe the robustness of reasoning processes against general perturbations that do not depend on the underlying data. **Statistical noise** is generated adversarially using Graph Neural Networks (GNNs), reflecting low-probability links that emerge from predictive uncertainty or bias in automated knowledge graph construction. Although synthetic, this form of noise models realistic mistakes produced by machine-learning systems, and mirrors the types of errors that error-detection models are typically asked to identify during KG construction. **Logical noise** captures violations of semantic constraints, such as disjointness axioms or domain and range restrictions. Because many real-world ontology errors stem from semantic confusion rather than random corruption, this type of noise directly stresses the logical structure of the ontology and provides a more targeted challenge to reasoning systems. By combining these three types of noise, we aim to approximate a spectrum of potential error

[*] https://www.cs.toronto.edu/~kriz/cifar.html

patterns observed in real-world ontology and knowledge graph construction, from accidental and statistically plausible mistakes to deliberate logical conflicts.

In this work, we explore the following research questions: how noise in ontologies can be characterized operationally, how controlled noise can be introduced into these structures, and how the impact of such perturbations can be evaluated in neurosymbolic reasoners. By exploring these questions, we aim to develop a framework for generating noisy benchmark datasets. This framework will facilitate the assessment of reasoners' robustness and effectiveness in handling noisy data, contributing toward more systematic evaluation practices in neurosymbolic AI (Singh, 2023b; Singh et al.).

We apply two neurosymbolic reasoners and one purely neural approach based on Graph Neural Networks to datasets with different noise levels to demonstrate their limitations in handling various types of noise. We then evaluate their performance under these noisy conditions. Many existing studies emphasize ontology completion or link prediction tasks, whereas our focus is on evaluating reasoning performance under noisy conditions. The goal of ontology/link completion is to discover plausible relations that complement the original ontology, as was the task performed in the work of Chen et al. (Chen et al., 2021). In contrast, our goal is to infer knowledge that logically follows from the given ontology.

The remainder of the paper is organized as follows: the existing literature on neurosymbolic ontology reasoners and benchmark datasets is reported in Related Work. Methodology presents the developed framework for injecting random, statistical, and logical noise. Experimental Setup presents the experimental setup. Results shows the results of the experiments, including performance metrics and analysis. Finally, Discussion discusses the strengths and limitations of the designed framework and explores potential extensions or improvements for future research, followed by Conclusion to conclude our work. The source code of the benchmark is available at `https://github.com/jloe2911/NoisyBench` under MIT License.

Related Work

Neurosymbolic approaches integrate diverse reasoning techniques, resulting in multiple variations in their evaluation. In Reasoning Techniques, we provide a brief overview of neurosymbolic reasoning methods that are used for our experiments, followed by a discussion of most commonly used benchmark datasets in Benchmark Datasets.

Reasoning Techniques

Henry Kautz, in his AAAI 2020 Robert S. Engelmore Memorial Award Lecture, discussed six categories of neurosymbolic AI systems as the "Future of AI" (Kautz, 2022). To showcase the variety in existing approaches, we categorize the reasoning methods used in our experiments into one of those categories. Henry Kautz defined the six categories of neurosymbolic AI identified as follows:

1. **Symbolic Neuro symbolic:** Symbolic data are embedded as vectors using methods such as Word2Vec (Mikolov et al., 2013), processed by neural networks, and subsequently converted back into symbolic representations.
2. **Symbolic[Neuro]:** A primarily symbolic framework that calls upon neural networks for specific tasks within a logical system; for example, in systems like AlphaGo (Silver et al., 2016), symbolic search (Monte Carlo Tree Search) queries neural networks to evaluate board states and prioritize moves.
3. **Neuro | Symbolic:** In this category, the neural network and symbolic module solve complementary tasks and communicate frequently to guide each other. This represents a refined integration of neural and symbolic approaches, exemplified by the Neurosymbolic Concept Learner (Mao et al., 2019).
4. **Neuro : Symbolic → Neuro:** Symbolic knowledge is used to constrain, supervise, or regularize the training and operation of neural networks, without necessarily producing explicit symbolic outputs. This is demonstrated in Deep Learning For Symbolic Mathematics (Lample and Charton, 2019).
5. **Neuro$_{\text{Symbolic}}$:** Logical rules or knowledge bases are directly embedded into neural architectures, shaping their internal representations and guiding generalization, as in the Logic Tensor Network (Badreddine et al., 2020).
6. **Neuro[Symbolic]:** Neural networks that are capable of logical reasoning at certain points during execution fall into this category.

In (Chen et al., 2021), the authors introduced *OWL2Vec**, which involves converting the symbolic input (i.e., ontologies and RDF graphs) to vectors, giving rise to *Symbolic Neuro symbolic*. The method leverages random walk and word embedding techniques to encode the semantics of OWL ontologies. Unlike traditional KG embedding methods, *OWL2Vec** considers not only the graph structure but also lexical information and logical constructors inherent in OWL ontologies. This comprehensive approach enables *OWL2Vec** to capture nuanced relationships between concepts, making it suitable for tasks requiring fine-grained reasoning, such as ontology completion and prediction. The empirical evaluation conducted with three real-world datasets, i.e., HeLis (Dragoni et al., 2018), FoodOn (Dooley et al., 2018) and Gene Ontology (GO) (Ashburner et al., 2000), demonstrates that *OWL2Vec** outperforms the state-of-the-art methods in class membership and class subsumption tasks. This suggests that *OWL2Vec** benefits from incorporating different aspects of ontology semantics, including graph structure, lexical information, and logical constructors.

In (Jackermeier et al., 2024), the authors proposed a novel ontology embedding method called *Box2EL* for `DL EL++`. The approach embeds symbolic reasoning inside neural engines, representing symbolic information in geometric or vector spaces and employing neural methods for reasoning tasks, resulting in the *Neuro[Symbolic]* category. Specifically, they addressed the challenge of ontology

completion in Description Logic (DL)-based OWL ontologies, which are widely used for knowledge representation. While classical deductive reasoning algorithms offer precise formal semantics for predicting missing facts in an ontology, recent years have seen a rise in interest in inductive reasoning techniques capable of deriving probable facts from an ontology. Inductive reasoning techniques, akin to those used in KG completion, involve learning ontology embeddings in a latent vector space while ensuring adherence to the semantics of the underlying DL. However, existing ontology embedding methods face shortcomings, particularly in faithfully modeling complex relations and role inclusion axioms, such as one-to-many, many-to-one, and many-to-many relations. This approach represents both concepts and roles as boxes (i.e., axis-aligned hyper-rectangles) and models inter-concept relationships using a bumping mechanism. The authors conduct an extensive experimental evaluation, achieving state-of-the-art results across a variety of datasets, i.e., GALEN (Rector, 2008), Gene Ontology (GO) (Ashburner et al., 2000) and Anatomy (a.k.a. Uberon) (Mungall et al., 2012), on the tasks of subsumption prediction, role assertion prediction and approximating deductive reasoning. Table 1 summarizes the main differences between *Box2EL* and *OWL2Vec**.

Table 1. Comparison of *Box2EL* and *OWL2Vec**.

	Box2EL	**OWL2Vec***
Core Idea	Embeds DL EL++ ontologies as **boxes** in vector space; **Neuro[Symbolic]** reasoning	Leverages ontologies/RDF graphs to generate vectors using **random walks + Word2Vec**; **Symbolic Neuro-Symbolic** reasoning
Ontology Input	EL++ fragment ontologies	Full OWL ontologies (graph, lexical, and logical constructors)
Embedding Type	Boxes/axis-aligned hyper-rectangles	Continuous vectors from word embeddings
Strengths	Faithful to DL semantics; supports reasoning over complex relations	Combines structure, lexical, and logical information; strong performance on multiple datasets
Weaknesses	Limited to EL++; requires DL-compliant ontologies	Embeddings approximate semantics; reasoning is probabilistic/pattern-based rather than formally guaranteed; embeddings are distributional, not geometric

Benchmark Datasets

There is a pressing need for standardized benchmark datasets for neurosymbolic reasoners to facilitate fair and consistent comparisons. Precisely, Singh et al. (Singh et al.) presented an overview of variations in neurosymbolic reasoning and evaluation approaches. Their overview reveals that similar works may differ significantly by employing distinct metrics and datasets to evaluate their

contributions. For instance, the works of Makni et al. (Makni and Hendler, 2019) and Ebrahimi et al. (Ebrahimi et al., 2021b) focus on RDFS entailment reasoning, aiming to replicate deductive reasoning processes. However, they adopt different metrics and datasets to assess the effectiveness and performance of their approaches. Such variations in evaluation criteria can lead to diverse insights and perspectives on the contributions within the field. Specifically, Makni et al. (Makni and Hendler, 2019) used LUBM and a scientist dataset derived from DBpedia as benchmarks, evaluating performance with Precision, Recall, and F1 score. In contrast, Ebrahimi et al. (Ebrahimi et al., 2021b) used LUBM and synthetic data, employing exact matching accuracy as their metric.

The existing traditional benchmarks such as LUBM (Lehigh University Benchmark) (Guo et al., 2005), UOBM (University Ontology Benchmark) (Ma et al., 2006) and OWL2Bench (Singh et al., 2020) lack suitability for evaluating neurosymbolic reasoners due to their narrow focus on conventional reasoning tasks. Traditional evaluations of reasoning systems often rely on metrics such as reasoning time, which may not align with the evaluation requirements of neurosymbolic reasoners. Although the ontologies of these benchmarks, along with those from the OWL Reasoner Evaluation (ORE) Competition (Parsia et al., 2017), can serve as initial datasets for neurosymbolic benchmarks, these datasets fall short of addressing the distinct challenges posed by neurosymbolic reasoning.

To our knowledge, no benchmarks or evaluation frameworks have been designed to evaluate and compare neurosymbolic reasoning systems. Most reasoner evaluations are performed on different publicly available ontologies, including but not restricted to SNOMED CT[†], Gene Ontology (GO) (Ashburner et al., 2000) and GALEN (Rector, 2008), as well as other ontologies available in public repositories such as DBpedia (Lehmann et al., 2014), YAGO (Suchanek et al., 2007), Wikidata (Vrandečić, 2012), Claros[‡], NCBO Bioportal[§] and AgroPortal[¶]. However, these offer a limited set of ontologies for evaluation, which does not cover the full spectrum of possible scenarios.

Methodology

This section outlines the mechanisms used in NSORN (Neurosymbolic Ontology Reasoning with Noise) to introduce noise into ontologies, specifically targeting the ABox, which contains instance-level information. We devised three distinct techniques to introduce noise into an ontology: logical (see Logical Noise), statistical (see Statistical Contradictions) and random noise (see Random Contradictions). Each method was designed to simulate a unique form of

[†] https://bioportal.bioontology.org/ontologies/SNOMEDCT
[‡] https://www.clarosnet.org
[§] https://bioportal.bioontology.org/
[¶] http://agroportal.lirmm.fr/

inconsistency or error, enabling us to assess the performance and robustness of ontology reasoning under various noisy conditions.

These techniques were specifically chosen to challenge the neurosymbolic reasoner's reasoning capabilities and to evaluate its resilience against varying levels and types of noise. By analyzing reasoning performance under such conditions, we can better understand the robustness and limitations of ontology-based systems.

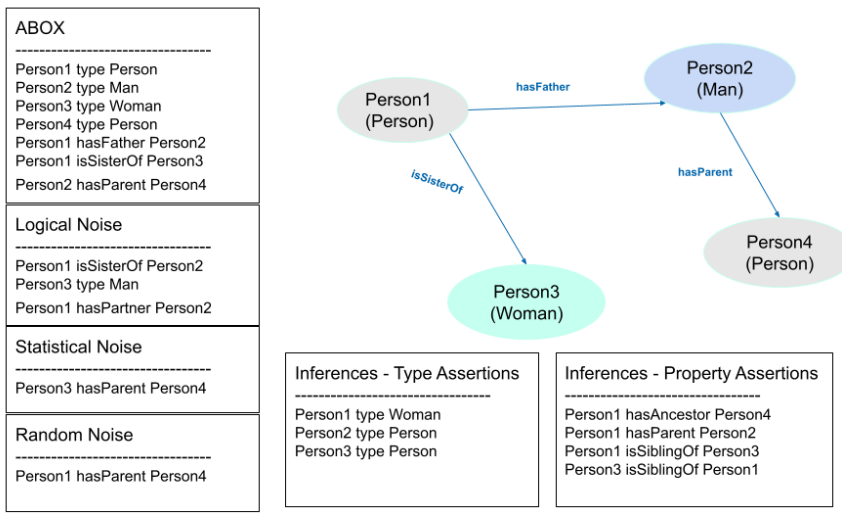

Figure 1. A toy example of a family ontology with possible inferences and different types of noise generation. The figure illustrates a sample ABox from the family ontology used in the benchmark. For this ABox, the possible type and property inferences are also provided. To observe the effect of different kinds of noise on these inferences, several noise types were introduced. For logical noise, since the assertion `Person1 hasFather Person2` exists in the ABox, an additional assertion `Person1 isSisterOf Person2` was added to invalidate it. The reason for adding this assertion is that there is a disjointness constraint between these two properties, which means that both these relations cannot simultaneously hold between the same pair of individuals. Similarly, for the assertion `Person3 type Woman`, the assertion `Person3 type Man` was added. This leads `Person3` to be both a `Woman` and a `Man`, resulting in a contradiction. As the last example of logical noise, a property is assigned to an individual that does not satisfy the property's domain or range restrictions. For instance, although the domain of `hasPartner` is `Marriage`, it was assigned with `Person1` as its subject, which leads to a domain/range violation. As an example of statistical noise, the least likely link between two entities is added—typically a relation between entities that normally have no connection at all. Finally, random noise is generated by creating an arbitrary relation between two randomly selected entities.

Logical Noise

Contradictions based on Disjoint Axioms This noise injection technique tests the robustness of reasoning engines by deliberately introducing contradictions into the ontology, evaluating their ability to handle inconsistencies. To introduce ABox noise within disjoint axioms (i.e., disjoint classes and disjoint object properties), we used the following approach:

1. **Extracting Disjoint Class Axioms:** We first identify all disjoint class axioms, denoted as $DisjointClasses(CE_1...CE_n)$, specifying that all class expressions CE_i $(1 \leq i \leq n)$ are pairwise disjoint[‖]. These axioms are used to generate noise that challenges the ontology's consistency.
2. **Introducing Noise:** To create inconsistencies, we assign k individuals to multiple disjoint classes CE_i and CE_j $(i \neq j)$. We first select existing individuals in the ontology; if additional examples are needed to reach the desired noise level, we introduce new, fictional individuals. For example, in Figure 1, if `Man` and `Woman` are disjoint, we assign `Person3` to both classes (`Person3 rdf:type Man` and `Person3 rdf:type Woman`). The parameter k controls the intensity of noise.

Similarly, we extract all disjoint object properties, denoted as $DisjointObjectProperties(OPE_1...OPE_n)$[**]. To introduce inconsistencies, we assign individuals to multiple disjoint properties OPE_i and OPE_j. Again, existing individuals are used first, and new individuals are added if necessary. For instance, if `hasFather` and `isSisterOf` are disjoint, an individual `Person1` is assigned both to `hasFather Person2` and `isSisterOf Person2`. By varying k, we can observe how different levels of noise affect reasoning performance, providing insights into system resilience and accuracy.

Contradictions based on Range/Domain Object properties in ontologies can have explicitly defined domains and ranges, which establish the types of individuals that are allowed to participate in a relationship. The domain specifies the class of individuals that can serve as the subject of the object property, while the range specifies the class of individuals that can serve as the object. Violations of these constraints lead to inconsistencies in the ontology, as they contradict the semantic rules established by the domain and range definitions.

For example, consider an object property `hasPartner` with a domain of `Marriage` and a range of `Person`. This means:

1. The subject of the `hasPartner` relationship must be a `Marriage`.
2. The object of the `hasPartner` relationship must be a `Person`.

[‖] https://www.w3.org/TR/owl2-syntax/#Disjoint_Classes
[**] https://www.w3.org/TR/owl2-syntax/#Disjoint_Object_Properties

If an assertion like `Person1 hasPartner Person2` is made, it would violate the domain constraint because `Person1` is not an instance of the class `Marriage`.

Such violations undermine the logical consistency of the ontology, making reasoning unreliable. Clearly defining and enforcing domain and range constraints ensures that the relationships in the ontology align with its intended semantics, enabling accurate reasoning and error detection.

Statistical Contradictions

We utilized Relational Graph Convolutional Networks (R-GCN) (Schlichtkrull et al., 2017) in our approach to model the complex relationships present in ontologies. R-GCN is particularly advantageous in handling multi-relational data as it extends the standard Graph Convolutional Network (GCN) (Kipf and Welling, 2016) by incorporating relation-specific transformations for edges. This allows the model to capture the semantics of different types of relationships in the graph.

We trained the R-GCN on a link prediction task, where the model predicts missing links based on existing data. After training, we identified the top k triples with lowest prediction scores, which were then added as noise to the ontology. Specifically, we altered existing triples—adding the modified versions as new triples—by replacing either the subject or the object with the entity that the R-GCN predicted to have the lowest probability. This method assesses the impact of noise generated through a statistical model and provides insights into the reasoner's handling of statistically improbable but plausible assertions.

Although statistical noise is generated probabilistically using a GNN, it is not explicitly constrained to satisfy ontology-level semantics, and may therefore occasionally introduce logical inconsistencies as an indirect effect.

Random Contradictions

We introduced k random triples into the ontology by corrupting either the subject or the object of existing triples and adding the resulting modified triples as new entries. This method simulates random noise and evaluates the reasoner's resilience to arbitrary disruptions in the data. Unlike previous noise injection techniques, this random approach contrasts the effects of systematic versus random noise on ontology reasoning. By corrupting existing triples, this method helps to understand how well the reasoner manages unexpected and non-systematic errors, crucial for assessing its robustness in real-world scenarios with unpredictable data inconsistencies.

Experimental Setup

Datasets

We used *Pizza* (Horridge, 2011), *Family* (Stevens and Stevens, 2008) and *OWL2Bench* (Singh et al., 2020) ontologies. Using the *Pizza* ontology, we created

an ABox generator to support experiments with synthetic data. The process for generating ABox data for the *Pizza* ontology begins by loading the *Pizza* TBox (Terminological Box) axioms. A custom instance generation step then automatically creates a specified number of individuals (ABox data), and their object properties are defined in a configuration. For this study, only `NamedPizza` class and `hasTopping` property are described in the configuration. Crucially, this generation leverages the TBox's inherent OWL restrictions (e.g., `only` or `some` constraints) to dynamically determine the appropriate target classes for object properties, thereby guaranteeing the generated ABox is semantically consistent with the ontology's definition. The final output is the complete ontology, comprising the original TBox and the newly populated ABox. In this study, we developed two datasets, *Pizza_100* and *Pizza_250*, comprising 100 and 250 individuals, respectively. The corresponding number of axioms for these datasets are 2,151 and 2,628, respectively.

Furthermore, this work incorporates the *Family* ontology, a well-known ontology designed to represent family relationships and genealogical information. The *Family* ontology provides a foundational framework for reasoning about kinship terms, familial roles, and relationships such as parent-child, sibling, and spouse connections. *Family* contains 2,516 axioms.

OWL2Bench encompasses a wide variety of axioms, including Class Expression Axioms, Object Property Axioms, Data Property Axioms, and Assertions. It serves as a benchmark to evaluate the coverage, scalability, and query performance of ontology reasoners across the four OWL 2 profiles: EL, QL, RL, and DL. *OWL2Bench* was developed as an extension of the well-known University Ontology Benchmark (UOBM), producing four distinct TBoxes—one for each OWL 2 profile. For this study, we used the TBox of *OWL2Bench1-DL*, expressed in the OWL "DL" profile. We generated a relatively smaller ABox than those in the *OWL2Bench* dataset as our work aims to assess the reasoning capabilities of neurosymbolic systems rather than their scalability. Our ABox generation for this ontology was restricted to the classes `University`, `Department`, `Person`, and `Course`, along with key object properties such as `hasDepartment`, `hasDoctoralDegreeFrom`, `teachesCourse`, and `takesCourse`. The final *OWL2Bench* ontology contains 1,841 axioms in total. Table 2 details the frequency of each axiom type within the dataset.

The majority of prior research has concentrated on ontology completion tasks (i.e., prediction) rather than on ontology reasoning tasks (i.e., inference) (Yang et al., 2025). Ontology or link completion involves identifying plausible relations that enrich the original ontology, as demonstrated in the study by Chen et al. (Chen et al., 2021). In prediction tasks, the training, validation, and testing datasets are typically created by randomly splitting the ontology axioms.

In contrast, our goal is to infer knowledge that logically follows from a given ontology. To this end, we adopt a method similar to that of Makni and Hendler (Makni and Hendler, 2019), following their approach to construct our training, test, and validation sets, thereby enabling proper ontology reasoning.

Table 2. Number of axioms in *Pizza_100*, *Pizza_250*, *Family* and *OWL2Bench*.

	Pizza_100	Pizza_250	Family	OWL2Bench
Class Expression Axioms				
Subclass Axioms	250	250	9	122
Equivalent Classes	13	13	5	19
Disjoint Classes	13	13	9	11
Object Property Axioms				
Object Subproperties	4	4	20	59
Equivalent Object Properties	0	0	1	4
Disjoint Object Properties	1	1	1	1
Inverse Object Properties	3	3	15	22
Object Property Domain	3	3	11	62
Object Property Range	4	4	13	57
Functional Object Properties	4	4	3	2
Inverse-Functional Object Properties	3	3	0	1
Reflexive Object Properties	0	0	0	1
Irreflexive Object Properties	0	0	0	2
Symmetric Object Properties	0	0	2	2
Asymmetric Object Properties	0	0	0	1
Transitive Object Properties	2	2	2	5
Role Composition	0	0	4	4
Data Property Axioms			0	
Data Subproperties	0	0	0	2
Equivalent Data Properties	0	0	0	1
Disjoint Data Properties	0	0	0	1
Data Property Domain	0	0	0	7
Data Property Range	0	0	0	1
Functional Data Properties	0	0	0	3
Assertions	0	0		
Individual Equality	0	0	0	0
Individual Inequality	0	0	1	0
Class Assertions	232	287	3	362
Positive Object Property Assertions	922	1,289	1,337	488
Negative Object Property Assertions	0	0	0	0
Positive Data Property Assertions	0	0	0	0
Negative Data Property Assertions	0	0	0	0

Let G denote the original ontology and I the ontology inferred using Pellet reasoner (Sirin et al., 2007). We use the Pellet reasoner to compute the complete set of logical inferences over the original ontology, which serves as the ground-truth reference for evaluating the inferred assertions produced by the evaluated systems. Since our approach is unsupervised, the graph G is ultimately added to G_{train}, while I is randomly assigned to G_{train}, G_{test} and G_{val}. The TBox is further added to G_{test} and G_{val}, ensuring that the reasoning tasks are based on a shared conceptual framework.

In practice, obtaining perfectly clean data is often impractical or costly, especially for ontologies derived from unstructured sources. Real-world datasets frequently contain errors, inconsistencies, and irrelevant information. To better reflect these conditions, we introduce noise to G_{test}. This enables us to rigorously evaluate the resilience of reasoners under realistic, noisy scenarios and to develop systems that are more robust to the imperfections commonly found in real-world data. Table 3 summarizes the number of triples, membership assertions, object property assertions, and remaining triples[††] for each dataset split across the four datasets used in this study.

Table 3. Summary of Triples, Membership, Object Property Assertions, and Remaining Triples for *Pizza_100*, *Pizza_250*, *Family* and *OWL2Bench*.

Dataset	Split	Triples	Membership	Object Property Assertions	Remaining Triples
Family	Train	137,856	1,694 (1.2%)	134,997 (97.9%)	1,165 (0.8%)
Family	Test	28,987	310 (1.1%)	28,532 (98.4%)	145 (0.5%)
Family	Val	28,984	309 (1.1%)	28,530 (98.4%)	145 (0.5%)
Pizza_100	Train	24,150	2,180 (9.0%)	20,478 (84.8%)	1,492 (6.2%)
Pizza_100	Test	4,701	364 (7.7%)	4,192 (89.2%)	145 (3.1%)
Pizza_100	Val	4,698	363 (7.7%)	4,190 (89.2%)	145 (3.1%)
Pizza_250	Train	30,413	2,637 (8.7%)	26,286 (86.4%)	1,490 (4.9%)
Pizza_250	Test	5,941	438 (7.4%)	5,358 (90.2%)	145 (2.4%)
Pizza_250	Val	5,938	437 (7.4%)	5,356 (90.2%)	145 (2.4%)
OWL2Bench	Train	4,123	2,049 (49.7%)	1,296 (31.4%)	778 (18.9%)
OWL2Bench	Test	1,462	510 (34.9%)	174 (11.9%)	778 (53.2%)
OWL2Bench	Val	1,459	508 (34.8%)	173 (11.9%)	778 (53.3%)

Metrics, Tasks and Reasoners

We used Mean Reciprocal Rank (MRR) and Hits@N to compare the performance of different neurosymbolic reasoners. MRR represents the average reciprocal rank, calculated by taking the reciprocal of the rank (1/rank) of the first relevant item retrieved. Hits@N measures the percentage of positive examples that appear in the top-k ranked predictions.

To assess how reasoners respond to noise, we focused on specific reasoning tasks: the first involves class assertions (also known as realization or membership), which determine whether an individual belongs to a specific class based on the logical definitions and constraints within the ontology, for example, `Alice rdf:type Person`. The second task involves object property assertions, that infer new relationships between two individuals in the ontology, such as `Alice hasSibling Bob`.

This experimental framework analyzes the impact of noise on reasoning outcomes, as well as to evaluate the performance and robustness of ontology

[††]Remaining triples refer to structural triples, e.g., those used for rdf:first/rdf:rest list encodings.

reasoning under different levels and types of noise. For our exploration into neurosymbolic reasoning, we have selected state-of-the-art neurosymbolic reasoners such as *Box2EL* (Jackermeier et al., 2024) and *OWL2Vec** (Chen et al., 2021). This work used the implementation of these methods provided by the mOWL library (Zhapa-Camacho et al., 2022). In addition, we employed a purely neural approach based on Relational Graph Convolutional Networks (R-GCN) (Schlichtkrull et al., 2017), implemented using the torch-geometric library (Fey and Lenssen, 2019).

Results

Family Results

Figure 2 summarizes the model performance on the *Family* ontology under various noise types and levels, evaluated on both Membership (M) and Object Property Assertion (OPA) tasks. Across all models, OPA performance remains consistently low. This is expected since 98.4% of the test triples correspond to object property assertion cases, many involving Pellet-generated inferences that require complex, non-local, multi-step reasoning—something non-symbolic reasoners models struggle to capture effectively.

For *OWL2Vec**, noise generally degrades performance. In the noise-free scenario, the model achieves membership and object property assertion MRRs of 0.562 and 0.040, respectively. As noise increases, performance declines steadily. Interestingly, a slight improvement in OPA occurs under GNN noise, likely because GNN-generated noisy triples preserve structural plausibility, reinforcing local relational patterns to some extent. The sharpest decline is observed at 100% corruption, where membership MRR drops to 0.166 (GNN noise) and OPA MRR to 0.028 (random noise).

Box2EL exhibits a similar downward trend. Starting from baseline MRRs of 0.299 (M) and 0.018 (OPA), its performance worsens with increasing noise, reaching a minimum membership MRR of 0.110 under 100% GNN noise and a near-zero OPA MRR of 0.001 under 100% logical noise.

In contrast, *R-GCN* demonstrates greater robustness and even some performance gains under certain noise conditions. In the noise-free setting, *R-GCN* scores MRRs of 0.323 (M) and 0.040 (OPA), with minimums of 0.295 (M) at 75% logical noise and 0.031 (OPA) at 100% logical noise. Because noise is introduced only in the test set, *R-GCN*'s robustness stems from its reliance on neighborhood aggregation patterns learned during training. Its predictions depend on structural consistency in node embeddings, allowing it to effectively identify noisy triples that deviate from expected patterns.

Pizza Results

Figure 3 shows the results for *Pizza* (100 individuals). *OWL2Vec** starts with membership and object property assertion MRRs near 0.28 but sharply declines

to 0.040 (M) and 0.004 (OPA) under 100% logical noise. *Box2EL* follows a similar trend, dropping from 0.114 (M) and 0.063 (OPA) to nearly zero under 100% logical noise. Once again, *R-GCN* proves more resilient, maintaining relatively stable performance.

Figure 4 shows the results for *Pizza* (250 individuals) and reflects consistent patterns. *OWL2Vec** performance decreases with increasing logical noise, with membership MRR falling from 0.254 to 0.040, and OPA MRR dropping from 0.280 to 0.003 at 100% logical noise. Again, the improvement in OPA observed under GNN noise during testing is likely because the GNN-generated noisy triples retain structural plausibility, which helps preserve local relational patterns to some extent. *Box2EL* steadily declines from MRRs of 0.114 (M) and 0.060 (OPA) to near-zero under 100% logical noise. *R-GCN* remains robust across noise types, with performance largely unaffected—mirroring the resilience observed before.

OWL2Bench Results

Figure 5 shows results on the *OWL2Bench* ontology. In the noise-free setting, *OWL2Vec** achieves the highest object property assertion scores (MRR around 0.7) and moderate membership MRR (0.28). While random and GNN noise degrade its performance gradually, logical noise has little impact on object property assertion accuracy and mainly affects membership predictions.

Box2EL performs significantly lower overall and suffers substantial performance losses under all noise types, with logical noise particularly devastating object property assertion accuracy, which falls near zero at moderate noise levels.

R-GCN offers a balanced and robust performance profile. Membership MRR remains relatively stable across noise types and levels, and object property assertion scores decline more gently than *Box2EL*'s under logical noise. Notably, *R-GCN* sometimes maintains or improves membership prediction under GNN noise, highlighting its effective use of graph structure to mitigate noise.

Overall Analysis

Across all ontologies and noise types tested, *R-GCN* consistently demonstrates superior robustness to noise compared to *OWL2Vec** and *Box2EL*. While all models experience declines in performance as noise increases—particularly on complex object property assertion tasks—*R-GCN*'s reliance on learned structural patterns and neighborhood aggregation enables it to better withstand noisy test data. *OWL2Vec** performs well in noise-free and logically noisy conditions for object property assertion but is more sensitive in membership tasks and under other noise types. Box2EL suffers the steepest performance drops, especially under logical noise, indicating limited noise tolerance.

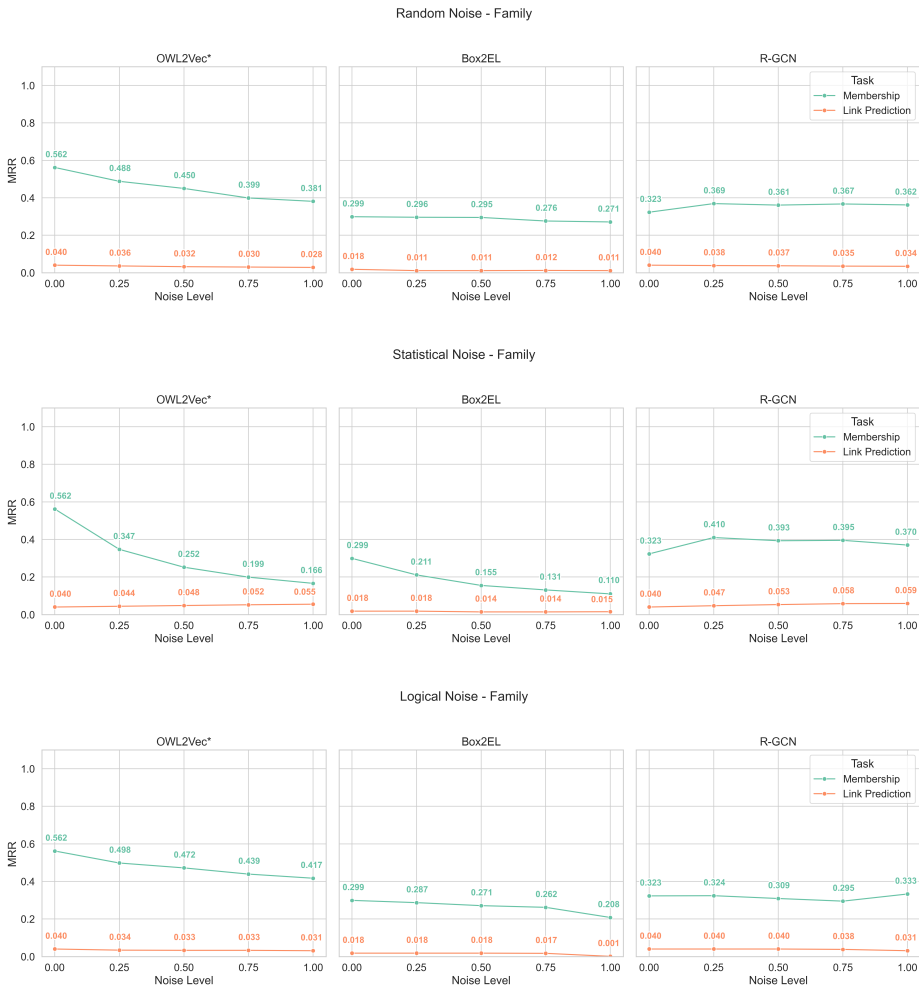

Figure 2. Results for *Family* ontology. x% noise indicates that the number of injected noisy assertions is x% of the original assertions.

Discussion

Our study investigates the application of noise injection methods to ontologies, examining their impact on various reasoning tasks. The proposed noise injection techniques are designed to be applicable across a wide range of ontologies. Based on our findings, we observed that across all ontologies, tasks, and noise types, *R-GCN* is consistently and substantially more robust to noisy test data than *OWL2Vec** and *Box2EL*. Its neighborhood-aggregation mechanism enables it to preserve performance even when a large portion of the test triples is

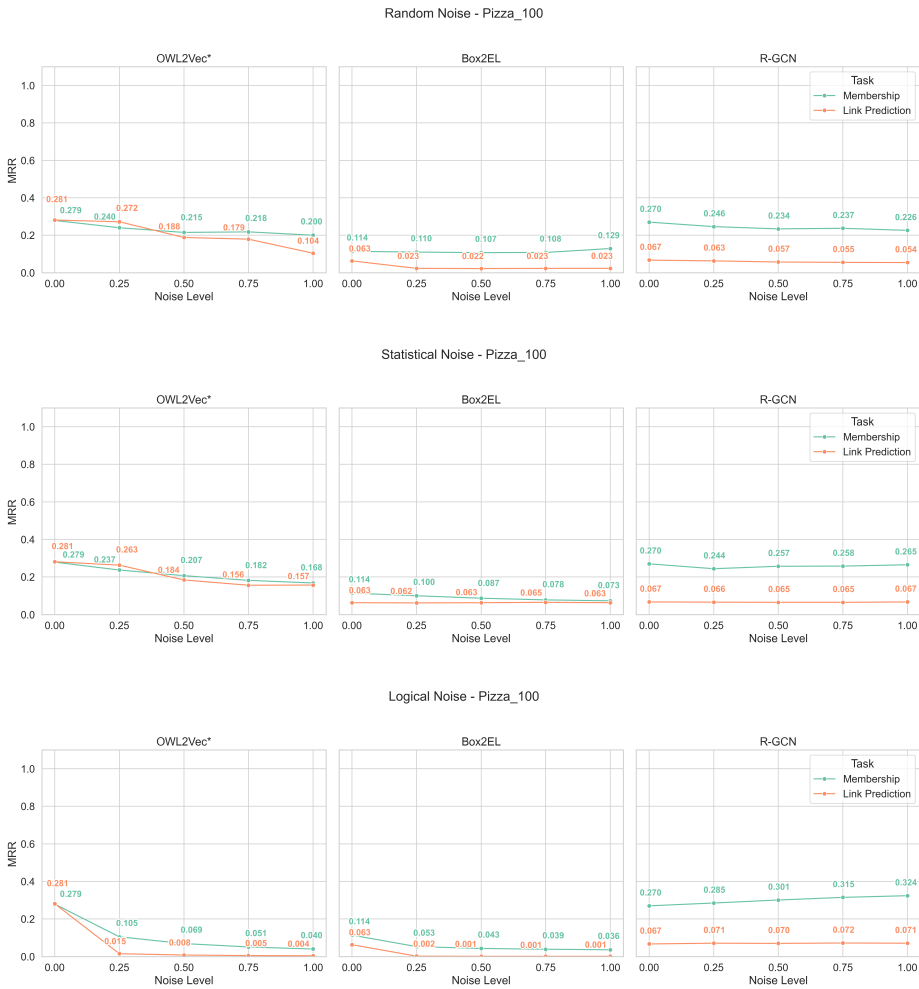

Figure 3. Results for *Pizza_100* ontology. x% noise indicates that the number of injected noisy assertions is x% of the original assertions.

corrupted, while embedding-based models—especially Box2EL—degrade sharply. This robustness holds for both membership prediction and object property assertion (OPA), the latter being particularly challenging due to the need for multi-step reasoning and the dominance of inference-based triples in the test sets.

These findings suggest that graph neural network-based approaches like *R-GCN* provide a more reliable foundation for reasoning over noisy, real-world ontologies, where data imperfections and complex inference paths are common.

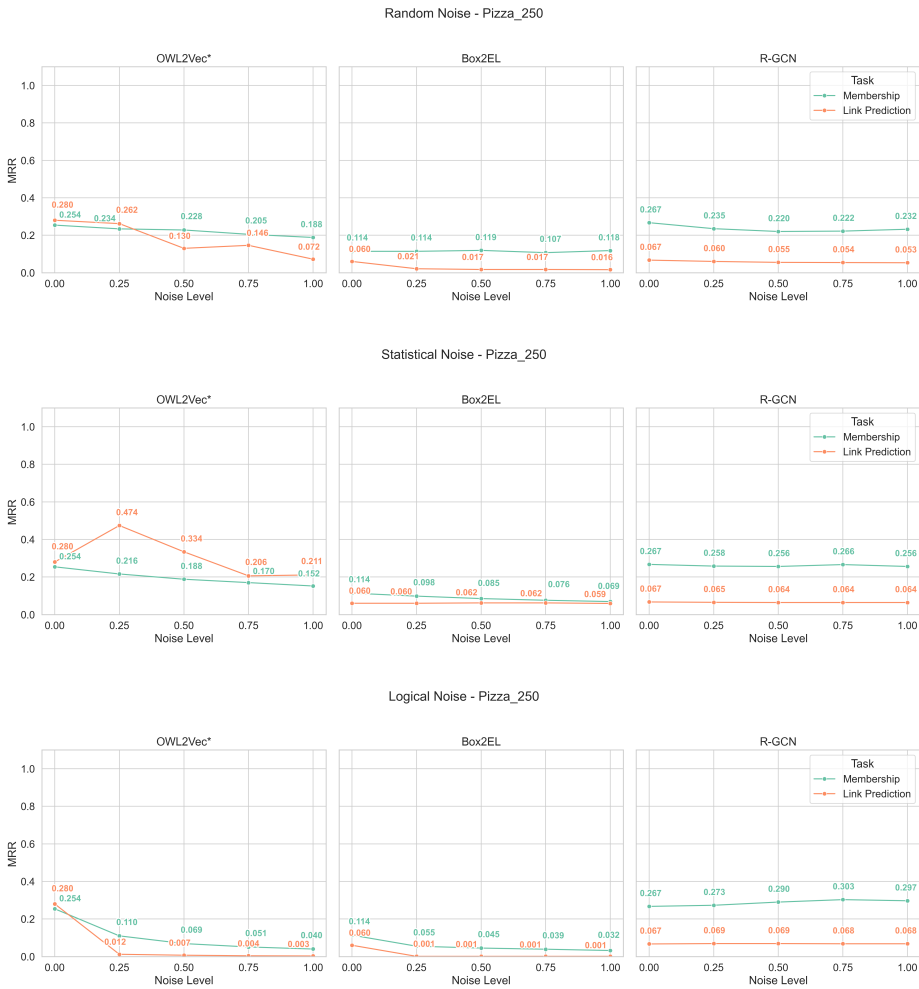

Figure 4. Results for *Pizza_250* ontology. x% noise indicates that the number of injected noisy assertions is x% of the original assertions.

However, the specific characteristics of each ontology significantly influence the effectiveness of noise injection, highlighting the need for tailored approaches in certain scenarios. For example, the specific relations in the test set may not effectively show the influence of noise introduced as these relations inherently resist noise. In *OWL2Bench*, `knows` relation is defined as reflexive (i.e., every individual 'knows' themselves), making it less sensitive to object property assertion inferences. These inferences hold regardless of corrupted assertions unless the TBox is modified. This raises questions about the validity of evaluating

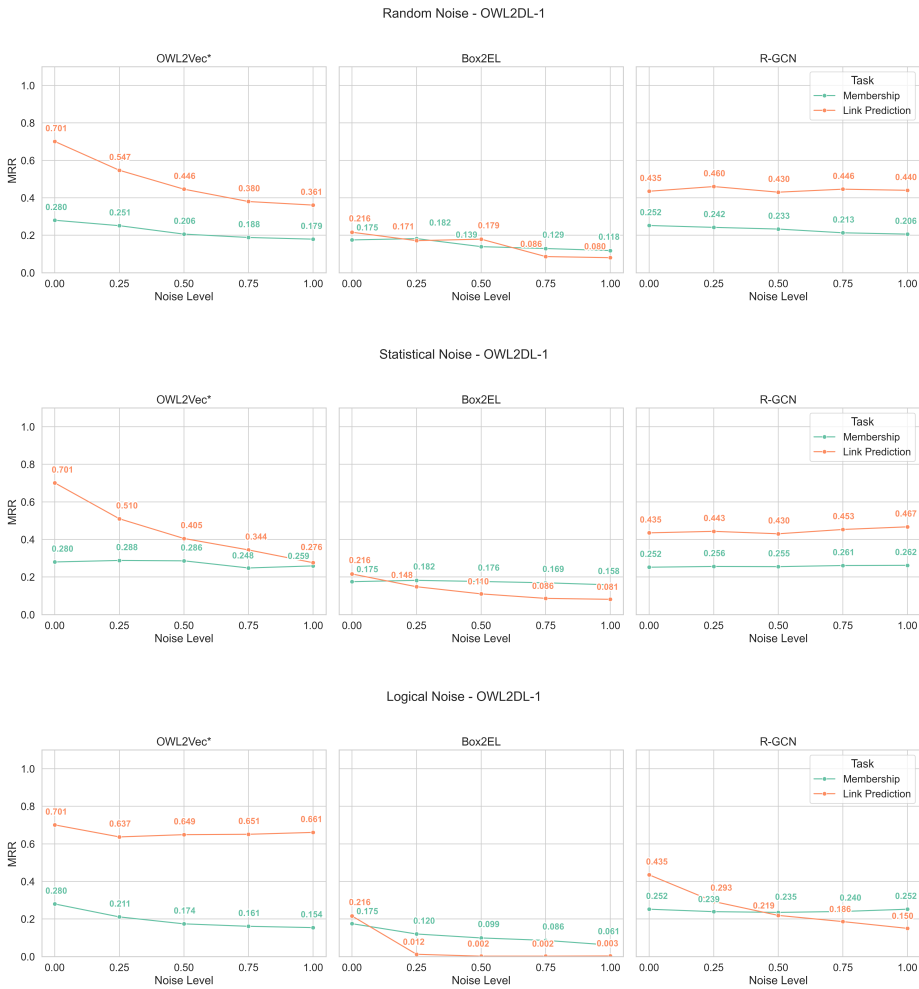

Figure 5. Results for *OWL2Bench* ontology. x% noise indicates that the number of injected noisy assertions is x% of the original assertions.

noise effects in scenarios where axiomatic properties dominate reasoning outcomes. Future work should consider refining testing sets or introducing variations in TBox definitions to better capture the influence of noise.

An important factor underlying the differing behaviors across ontologies is the composition of the test sets, particularly the ratio of membership versus object property assertion triples. As shown in Table 3, these ratios vary dramatically between ontologies, which in turn affects how strongly noise impacts model performance. For example, in the *Family* ontology, OPA triples constitute over

98% of the test set. Since many of these OPAs arise from multi-step inferences produced by Pellet, the test set is dominated by structurally complex, inference-heavy triples—making it inherently challenging and causing all models to exhibit low OPA performance even without noise. In this setting, noise injected into test triples disproportionately affects OPAs, amplifying the performance gap between *R-GCN* and embedding-based models.

In contrast, *OWL2Bench* has a much more balanced distribution, with only around 12% OPA triples and a substantially larger proportion of membership triples. This composition makes the test set less sensitive to inference-heavy corruption, which helps explain why *OWL2Vec** attains relatively strong OPA performance and why noise in OPAs produces less dramatic declines compared with the *Family* ontology. The *Pizza* ontologies occupy an intermediate position, with 85–90% OPA triples, resulting in noise effects that are more severe than *OWL2Bench* but less extreme than *Family*. The relative stability of *R-GCN* across these datasets stems from its reliance on learned neighborhood structure rather than isolated triple accuracy, allowing it to generalize across ontologies despite their differing test-set compositions.

These observations highlight that robustness cannot be interpreted independently of the structural properties of the evaluation set. Ontologies whose test splits are dominated by complex inference-derived triples amplify the impact of noisy test data, disproportionately penalizing embedding-based models. Conversely, datasets with more balanced triple ratios may under-reflect the challenges posed by noise. Future benchmarks should therefore consider standardizing or controlling test-set composition when evaluating noise robustness, or at minimum report such statistics to contextualize cross-ontology comparisons.

Furthermore, it should be noted that the results from previous works, such as the work of (Zhapa-Camacho and Hoehndorf, 2023), are not comparable to ours because our proposed benchmark focuses on evaluating ontology reasoning rather than ontology completion. Ontology reasoning refers to inferring logically consistent relationships from existing data and rules, which is inherently more complex. This complexity arises because reasoning requires the system to consider all possible logical implications of the data, making it more sensitive to inconsistencies and noise in the dataset. Consequently, the metrics may reflect this added difficulty, leading to poorer results compared to approaches that focus solely on completing the ontology.

While our initial exploration centered on introducing noise through the addition of logical contradictions or corruption of triples with low probability of occurrence, many other types of axioms and noise patterns merit investigation. Future research could involve examining various inconsistencies, contradictions, and errors that frequently occur in real-world ontologies, thereby enhancing the diversity of noise generation techniques. In particular, introducing noise in the TBox (e.g., modifying class hierarchies, altering domain and range constraints, or introducing invalid equivalence axioms) could offer valuable insights into how structural

and logical inconsistencies impact reasoning outcomes. Furthermore, future work could focus on establishing standardized metrics and evaluation frameworks to consistently measure the performance of neurosymbolic reasoning systems.

Conclusion

This paper presents NSORN (Neurosymbolic Ontology Reasoning with Noise), a reproducible and extensible framework for generating noisy benchmark datasets, with a particular focus on controlled perturbations of ABox assertions. We developed three techniques for introducing noise into the ABox: logical noise, statistical noise, and random noise. Logical noise is introduced by contradicting disjoint axioms or violating domain/range constraints of object properties. Statistical noise, on the other hand, is introduced using Graph Neural Networks, which add new links with low probability scores to mimic biases commonly found in automated KG construction pipelines. Random noise involves unpredictable, arbitrarily alternations to ABox assertions. These methods were designed to evaluate the robustness and performance of ontology-based neurosymbolic reasoners under various noise conditions.

We evaluated the performance of existing neurosymbolic reasoners on *Pizza*, *Family* and *OWL2Bench* under various noise types and levels. The resulting benchmarks were tested on state-of-the-art neurosymbolic reasoners, *Box2EL* and *OWL2Vec**, as well as a purely neural method, *R-GCN*. The reasoning tasks considered include membership and object property assertions, with the aim of evaluating how effectively these reasoners handle noise. Our experimental results show how different modeling paradigms respond to varying types and levels of noise. In our experimental setup, the GNN-based model (R-GCN) exhibited greater empirical robustness to injected perturbations, particularly under logically inconsistent noise, while embedding-based neurosymbolic models showed more pronounced degradation. In contrast to many prior studies that focus primarily on ontology completion tasks, our emphasis is on ontology reasoning under noisy conditions, where inference quality is directly evaluated. The source code of NSORN is available at https://github.com/jloe2911/NoisyBench under MIT License.

Gunjan Singh and Raghava Mutharaju would like to acknowledge the partial support of the Infosys Centre for Artificial Intelligence (CAI), IIIT-Delhi, India, in this work.

References

Anke LE, Declerck T, Gromann D, Makni B, Hendler J, Gromann D, Espinosa Anke L and Declerck T (2019) Deep learning for noise-tolerant rdfs reasoning. *Semant. Web* 10(5): 823–862. DOI:10.3233/SW-190363. URL https://doi.org/10.3233/SW-190363.

Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM and Sherlock G (2000) Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat Genet* 25(1): 25–29. DOI:10.1038/75556. URL http://www.ncbi.nlm.nih.gov/pubmed/10802651.

Badreddine S, d'Avila Garcez AS, Serafini L and Spranger M (2020) Logic tensor networks. *CoRR* abs/2012.13635. URL https://arxiv.org/abs/2012.13635.

Banerjee D, Usbeck R, Mihindukulasooriya N, Jaradeh MY, Auer S, Singh G, Mutharaju R and Kapanipathi P (eds.) (2023) *Joint Proceedings of Scholarly QALD 2023 and SemREC 2023*, number 3592 in CEUR Workshop Proceedings. Aachen. URL http://ceur-ws.org/Vol-3592/.

Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J and Zaremba W (2016) Openai gym.

Chen J, Hu P, Jimenez-Ruiz E, Holter OM, Antonyrajah D and Horrocks I (2021) Owl2vec*: Embedding of owl ontologies.

Deng L (2012) The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine* 29(6): 141–142.

Dooley DM, Griffiths EJ, Gosal GS, Buttigieg PL, Hoehndorf R, Lange MC, Schriml LM, Brinkman FSL and Hsiao WWL (2018) Foodon: a harmonized food ontology to increase global food traceability, quality control and data integration. *npj Science of Food* 2(1): 23–. DOI:10.1038/s41538-018-0032-6. URL https://doi.org/10.1038/s41538-018-0032-6.

Dragoni M, Bailoni T, Maimone R and Eccher C (2018) Helis: An ontology for supporting healthy lifestyles. In: Vrandečić D, Bontcheva K, Suárez-Figueroa MC, Presutti V, Celino I, Sabou M, Kaffee LA and Simperl E (eds.) *The Semantic Web – ISWC 2018*. Cham: Springer International Publishing. ISBN 978-3-030-00668-6, pp. 53–69.

Ebrahimi M, Eberhart A, Bianchi F and Hitzler P (2021a) Towards bridging the neuro-symbolic gap: deep deductive reasoners. *Applied Intelligence* 51(9): 6326–6348. DOI:10.1007/s10489-020-02165-6. URL https://doi.org/10.1007/s10489-020-02165-6.

Ebrahimi M, Eberhart A and Hitzler P (2021b) On the Capabilities of Pointer Networks for Deep Deductive Reasoning. *CoRR* abs/2106.09225. URL https://arxiv.org/abs/2106.09225.

Fey M and Lenssen JE (2019) Fast graph representation learning with pytorch geometric. URL https://arxiv.org/abs/1903.02428.

Garcez Ad, Besold TR, De Raedt L, Földiak P, Hitzler P, Icard T, Kühnberger KU, Lamb LC, Miikkulainen R and Silver DL (2015) Neural-symbolic learning and reasoning: contributions and challenges. In: *2015 AAAI Spring Symposium Series*.

Gruber TR (1993) A translation approach to portable ontology specifications. *Knowledge acquisition* 5(2): 199–220.

Guo Y, Pan Z and Heflin J (2005) LUBM: A Benchmark for OWL Knowledge Base Systems. *Journal of Web Semantics.* 3(2-3): 158–182.

Horridge M (2011) The pizza ontology. URL https://protege.stanford.edu/publications/ontology/pizza.owl. From the Protege OWL Tutorial, University of Manchester.

Jackermeier M, Chen J and Horrocks I (2024) Dual box embeddings for the description logic el++.

Kautz HA (2022) The third AI summer: AAAI robert s. engelmore memorial lecture. *AI Magazine* 43(1): 93–104. DOI:10.1609/aimag.v43i1.19122. URL https://doi.org/10.1609/aimag.v43i1.19122.

Kipf TN and Welling M (2016) Semi-supervised classification with graph convolutional networks. *CoRR* abs/1609.02907. URL http://arxiv.org/abs/1609.02907.

Lample G and Charton F (2019) Deep learning for symbolic mathematics. URL https://arxiv.org/abs/1912.01412.

Lehmann J, Isele R, Jakob M, Jentzsch A, Kontokostas D, Mendes P, Hellmann S, Morsey M, Van Kleef P, Auer S and Bizer C (2014) Dbpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal* 6. DOI:10.3233/SW-140134.

Liu X, Liu Y and Hu W (2024) Knowledge graph error detection with contrastive confidence adaption. In: *Proceedings of the AAAI conference on artificial intelligence*, volume 38. pp. 8824–8831.

Ma L, Yang Y, Qiu G Z and Xie, Pan Y and Liu S (2006) Towards a Complete OWL Ontology Benchmark. In: *The Semantic Web: Research and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 125–139.

Makni B, Ebrahimi M, Gromann D and Eberhart A (2021) Neuro-symbolic semantic reasoning. In: Hitzler P and Sarker MK (eds.) *Neuro-Symbolic Artificial Intelligence: The State of the Art*, *Frontiers in Artificial Intelligence and Applications*, volume 342. IOS Press. ISBN 978-1-64368-244-0, pp. 253–279. DOI:10.3233/FAIA210358. URL https://doi.org/10.3233/FAIA210358.

Makni B and Hendler JA (2019) Deep learning for noise-tolerant RDFS reasoning. *Semantic Web* 10(5): 823–862. DOI:10.3233/SW-190363. URL https://doi.org/10.3233/SW-190363.

Mao J, Gan C, Kohli P, Tenenbaum JB and Wu J (2019) The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *CoRR* abs/1904.12584. URL http://arxiv.org/abs/1904.12584.

Mikolov T, Chen K, Corrado GS and Dean J (2013) Efficient estimation of word representations in vector space. In: *International Conference on Learning Representations*. URL https://api.semanticscholar.org/CorpusID:5959482.

Mungall CJ, Torniai C, Gkoutos GV, Lewis SE and Haendel MA (2012) Uberon, an integrative multi-species anatomy ontology. *GenomeBiology.com* 13(1). DOI:10.1186/gb-2012-13-1-r5.

Parsia B, Matentzoglu N, Gonçalves RS, Glimm B and Steigmiller A (2017) The OWL Reasoner Evaluation (ORE) 2015 Competition Report. *Journal of Automated Reasoning* 59(4): 455–482. DOI:10.1007/s10817-017-9406-8. URL https://doi.org/10.1007/s10817-017-9406-8.

Raji ID, Bender EM, Paullada A, Denton E and Hanna A (2021) AI and the everything in the whole wide world benchmark. *CoRR* abs/2111.15366. URL https://arxiv.org/abs/2111.15366.

Rector AL (2008) The galen high level ontology. URL https://api.semanticscholar.org/CorpusID:73607647.

Sarker MK, Zhou L, Eberhart A and Hitzler P (2021) Neuro-symbolic artificial intelligence. *AI Communications* 34(3): 197–209.

Schlichtkrull M, Kipf TN, Bloem P, van den Berg R, Titov I and Welling M (2017) Modeling relational data with graph convolutional networks.

Sheth A, Roy K and Gaur M (2023) Neurosymbolic artificial intelligence (why, what, and how). *IEEE Intelligent Systems* 38(3): 56–62. DOI:10.1109/MIS.2023.3268724.

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap TP, Leach M, Kavukcuoglu K, Graepel T and Hassabis D (2016) Mastering the game of Go with deep neural networks and tree search. *Nat.* 529(7587): 484–489. DOI:10.1038/nature16961. URL https://doi.org/10.1038/nature16961.

Singh G (2023a) Benchmarking symbolic and neuro-symbolic description logic reasoners. Doctoral Consortium at International Semantic Web Conference.

Singh G (2023b) Benchmarking symbolic and neuro-symbolic description logic reasoners. In: d'Amato C and Pan JZ (eds.) *Proceedings of the Doctoral Consortium at ISWC 2023 co-located with 22nd International Semantic Web Conference (ISWC 2023), Athens, Greece, November 7, 2023*, *CEUR Workshop Proceedings*, volume 3678. CEUR-WS.org. URL https://ceur-ws.org/Vol-3678/paper11.pdf.

Singh G, Bhatia S and Mutharaju R (2020) OWL2Bench: A Benchmark for OWL 2 Reasoners. In: Pan JZ, Tamma VAM, d'Amato C, Janowicz K, Fu B, Polleres A, Seneviratne O and Kagal L (eds.) *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II*, *Lecture Notes in Computer Science*, volume 12507. Springer, pp. 81–96. DOI:10.1007/978-3-030-62466-8_6. URL https://doi.org/10.1007/978-3-030-62466-8_6.

Singh G, Mutharaju R, Bhatia S and Tommasini R (????) Benchmarking neuro-symbolic reasoners: Existing challenges and a way forward. URL https://neurosymbolic-ai-journal.com/system/files/nai-paper-774_0.pdf.

Singh G, Mutharaju R and Kapanipathi P (eds.) (2021) *Proceedings of the Semantic Reasoning Evaluation Challenge (SemREC 2021)*, number 3123 in CEUR Workshop Proceedings. Aachen. URL http://ceur-ws.org/Vol-3123/.

Singh G, Mutharaju R, Kapanipathi P, Mihindukulasooriya N, Dubey M, Usbeck R and Banerjee D (eds.) (2022) *Joint Proceedings of SemREC 2022 and SMART 2022*, number 3337 in CEUR Workshop Proceedings. Aachen. URL http://ceur-ws.org/Vol-3337/.

Sirin E, Parsia B, Grau BC, Kalyanpur A and Katz Y (2007) Pellet: A practical owl-dl reasoner. *Journal of Web Semantics* 5(2): 51–53. DOI:https://doi.org/10.1016/j.websem.2007.03.004. URL https://www.sciencedirect.com/science/article/pii/S1570826807000169. Software Engineering and the Semantic Web.

Stevens R and Stevens M (2008) A family history knowledge base using owl 2. In: *OWL: Experiences and Directions*. URL https://api.semanticscholar.org/CorpusID:10478581.

Suchanek FM, Kasneci G and Weikum G (2007) YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In: *Proceedings of the 16th International Conference on World Wide Web*, WWW '07. New York, NY, USA: Association for Computing Machinery. ISBN 9781595936547, p. 697–706.

DOI:10.1145/1242572.1242667. URL https://doi.org/10.1145/1242572.1242667.

Vrandečić D (2012) Wikidata: A new platform for collaborative data collection. In: *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion. New York, NY, USA: Association for Computing Machinery. ISBN 9781450312301, p. 1063–1064. DOI:10.1145/2187980.2188242. URL https://doi.org/10.1145/2187980.2188242.

Wang A, Singh A, Michael J, Hill F, Levy O and Bowman SR (2019) Glue: A multi-task benchmark and analysis platform for natural language understanding.

Yang H, Chen J, He Y, Gao Y and Horrocks I (2025) Language models as ontology encoders. URL https://arxiv.org/abs/2507.14334.

Yu D, Yang B, Liu D, Wang H and Pan S (2023) A survey on neural-symbolic learning systems. *Neural Networks* 166: 105–126. DOI:https://doi.org/10.1016/j.neunet.2023.06.028. URL https://www.sciencedirect.com/science/article/pii/S0893608023003398.

Zhapa-Camacho F and Hoehndorf R (2023) Evaluating different methods for semantic reasoning over ontologies. In: *QALD/SemREC@ ISWC*.

Zhapa-Camacho F, Kulmanov M and Hoehndorf R (2022) mOWL: Python library for machine learning with biomedical ontologies. *Bioinformatics* DOI:10.1093/bioinformatics/btac811. URL https://doi.org/10.1093/bioinformatics/btac811. Btac811.

Supporting material

Note: Bolded values indicate the lowest MRR for each reasoning task, highlighting the conditions under which each reasoner performs least effectively.

Table 4. Results on *family* ontology using *OWL2Vec** reasoner.

		MRR	**Hits@1**	**Hits@5**	**Hits@10**
No Noise	Membership	0.562	0.621	1.000	1.000
	Object Property Assertion	0.040	0.027	0.059	0.086
25% Random Noise	Membership	0.488	0.546	0.882	0.885
	Object Property Assertion	0.036	0.023	0.050	0.074
50% Random Noise	Membership	0.450	0.505	0.808	0.813
	Object Property Assertion	0.032	0.020	0.045	0.066
75% Random Noise	Membership	0.399	0.437	0.718	0.725
	Object Property Assertion	0.030	0.018	0.041	0.061
100% Random Noise	Membership	0.381	0.410	0.698	0.708
	Object Property Assertion	**0.028**	0.016	0.037	0.057
25% GNN Noise	Membership	0.347	0.383	0.615	0.621
	Object Property Assertion	0.044	0.032	0.059	0.083
50% GNN Noise	Membership	0.252	0.276	0.444	0.449
	Object Property Assertion	0.048	0.036	0.060	0.081
75% GNN Noise	Membership	0.199	0.216	0.350	0.353
	Object Property Assertion	0.052	0.040	0.062	0.081
100% GNN Noise	Membership	**0.166**	0.182	0.291	0.294
	Object Property Assertion	0.055	0.044	0.063	0.081
25% Logical Noise	Membership	0.498	0.528	0.950	1.000
	Object Property Assertion	0.034	0.021	0.047	0.070
50% Logical Noise	Membership	0.472	0.478	0.939	1.000
	Object Property Assertion	0.033	0.020	0.046	0.069
75% Logical Noise	Membership	0.439	0.424	0.912	1.000
	Object Property Assertion	0.033	0.020	0.046	0.069
100% Logical Noise	Membership	0.417	0.301	0.727	1.000
	Object Property Assertion	0.031	0.019	0.042	0.062

Table 5. Results on *family* ontology using *Box2EL* reasoner.

		MRR	Hits@1	Hits@5	Hits@10
No Noise	Membership	0.299	0.165	0.882	1.000
	Object Property Assertion	0.018	0.007	0.019	0.033
25% Random Noise	Membership	0.296	0.204	0.793	0.826
	Object Property Assertion	0.011	0.003	0.011	0.019
50% Random Noise	Membership	0.295	0.224	0.728	0.760
	Object Property Assertion	0.011	0.003	0.009	0.018
75% Random Noise	Membership	0.276	0.210	0.664	0.694
	Object Property Assertion	0.012	0.003	0.010	0.020
100% Random Noise	Membership	0.271	0.209	0.643	0.675
	Object Property Assertion	0.011	0.003	0.009	0.018
25% GNN Noise	Membership	0.211	0.140	0.576	0.602
	Object Property Assertion	0.018	0.007	0.018	0.032
50% GNN Noise	Membership	0.155	0.100	0.414	0.430
	Object Property Assertion	0.014	0.004	0.012	0.023
75% GNN Noise	Membership	0.131	0.091	0.333	0.344
	Object Property Assertion	0.014	0.004	0.013	0.023
100% GNN Noise	Membership	**0.110**	0.078	0.273	0.284
	Object Property Assertion	0.015	0.005	0.014	0.025
25% Logical Noise	Membership	0.287	0.164	0.836	0.908
	Object Property Assertion	0.018	0.007	0.019	0.032
50% Logical Noise	Membership	0.271	0.143	0.803	0.893
	Object Property Assertion	0.018	0.006	0.018	0.031
75% Logical Noise	Membership	0.262	0.128	0.773	0.867
	Object Property Assertion	0.017	0.006	0.018	0.032
100% Logical Noise	Membership	0.208	0.125	0.416	0.614
	Object Property Assertion	**0.001**	0.000	0.000	0.000

Table 6. Results on *family* ontology using *R-GCN*.

		MRR	**Hits@1**	**Hits@5**	**Hits@10**
No Noise	Membership	0.323	0.054	0.973	0.998
	Object Property Assertion	0.040	0.016	0.054	0.080
25% Random Noise	Membership	0.369	0.111	0.924	0.973
	Object Property Assertion	0.038	0.014	0.051	0.077
50% Random Noise	Membership	0.361	0.110	0.883	0.942
	Object Property Assertion	0.037	0.014	0.049	0.074
75% Random Noise	Membership	0.367	0.123	0.842	0.926
	Object Property Assertion	0.035	0.013	0.046	0.071
100% Random Noise	Membership	0.362	0.125	0.816	0.913
	Object Property Assertion	0.034	0.012	0.044	0.068
25% GNN Noise	Membership	0.410	0.157	0.889	0.990
	Object Property Assertion	0.047	0.018	0.066	0.097
50% GNN Noise	Membership	0.393	0.151	0.793	0.982
	Object Property Assertion	0.053	0.021	0.075	0.110
75% GNN Noise	Membership	0.395	0.171	0.734	0.975
	Object Property Assertion	0.058	0.024	0.085	0.123
100% GNN Noise	Membership	0.370	0.149	0.704	0.929
	Object Property Assertion	0.059	0.022	0.086	0.128
25% Logical Noise	Membership	0.324	0.054	0.934	0.995
	Object Property Assertion	0.040	0.015	0.053	0.080
50% Logical Noise	Membership	0.309	0.046	0.919	0.995
	Object Property Assertion	0.040	0.015	0.053	0.079
75% Logical Noise	Membership	**0.295**	0.036	0.884	0.996
	Object Property Assertion	0.038	0.014	0.051	0.077
100% Logical Noise	Membership	0.333	0.106	0.794	0.991
	Object Property Assertion	**0.031**	0.009	0.036	0.063

Table 7. Results on *pizza_100* ontology using *OWL2Vec** reasoner.

		MRR	Hits@1	Hits@5	Hits@10
No Noise	Membership	0.279	0.176	0.616	0.980
	Object Property Assertion	0.281	0.278	0.278	0.278
25% Random Noise	Membership	0.240	0.158	0.497	0.854
	Object Property Assertion	0.272	0.269	0.269	0.269
50% Random Noise	Membership	0.215	0.142	0.431	0.770
	Object Property Assertion	0.188	0.180	0.205	0.220
75% Random Noise	Membership	0.218	0.151	0.434	0.717
	Object Property Assertion	0.179	0.174	0.174	0.174
100% Random Noise	Membership	0.200	0.125	0.416	0.716
	Object Property Assertion	0.104	0.097	0.097	0.097
25% GNN Noise	Membership	0.237	0.144	0.510	0.855
	Object Property Assertion	0.263	0.246	0.353	0.387
50% GNN Noise	Membership	0.207	0.125	0.452	0.749
	Object Property Assertion	0.184	0.171	0.244	0.290
75% GNN Noise	Membership	0.182	0.110	0.392	0.658
	Object Property Assertion	0.156	0.140	0.206	0.257
100% GNN Noise	Membership	0.168	0.106	0.340	0.604
	Object Property Assertion	0.157	0.131	0.199	0.255
25% Logical Noise	Membership	0.105	0.071	0.209	0.325
	Object Property Assertion	0.015	0.012	0.013	0.016
50% Logical Noise	Membership	0.069	0.046	0.124	0.194
	Object Property Assertion	0.008	0.006	0.007	0.009
75% Logical Noise	Membership	0.051	0.031	0.091	0.140
	Object Property Assertion	0.005	0.004	0.005	0.006
100% Logical Noise	Membership	**0.040**	0.023	0.066	0.108
	Object Property Assertion	**0.004**	0.003	0.004	0.005

Table 8. Results on *pizza_100* ontology using *Box2EL* reasoner.

		MRR	**Hits@1**	**Hits@5**	**Hits@10**
No Noise	Membership	0.114	0.001	0.083	0.903
	Object Property Assertion	0.063	0.026	0.080	0.155
25% Random Noise	Membership	0.110	0.015	0.098	0.758
	Object Property Assertion	0.023	0.009	0.024	0.042
50% Random Noise	Membership	0.107	0.021	0.113	0.666
	Object Property Assertion	0.022	0.008	0.021	0.039
75% Random Noise	Membership	0.108	0.029	0.123	0.624
	Object Property Assertion	0.023	0.008	0.025	0.046
100% Random Noise	Membership	0.129	0.052	0.181	0.632
	Object Property Assertion	0.023	0.008	0.025	0.043
25% GNN Noise	Membership	0.100	0.001	0.066	0.763
	Object Property Assertion	0.062	0.025	0.080	0.153
50% GNN Noise	Membership	0.087	0.000	0.064	0.653
	Object Property Assertion	0.063	0.025	0.082	0.155
75% GNN Noise	Membership	0.078	0.003	0.051	0.557
	Object Property Assertion	0.065	0.028	0.084	0.152
100% GNN Noise	Membership	0.073	0.003	0.051	0.517
	Object Property Assertion	0.063	0.026	0.081	0.152
25% Logical Noise	Membership	0.053	0.004	0.044	0.270
	Object Property Assertion	0.002	0.000	0.000	0.001
50% Logical Noise	Membership	0.043	0.007	0.044	0.151
	Object Property Assertion	0.001	0.000	0.000	0.000
75% Logical Noise	Membership	0.039	0.007	0.040	0.105
	Object Property Assertion	0.001	0.000	0.000	0.000
100% Logical Noise	Membership	**0.036**	0.008	0.034	0.080
	Object Property Assertion	**0.001**	0.000	0.000	0.000

Table 9. Results on *pizza_100* ontology using *R-GCN*.

		MRR	Hits@1	Hits@5	Hits@10
No Noise	Membership	0.270	0.137	0.326	0.778
	Object Property Assertion	0.067	0.014	0.072	0.149
25% Random Noise	Membership	0.246	0.121	0.276	0.735
	Object Property Assertion	0.063	0.013	0.067	0.136
50% Random Noise	Membership	0.234	0.118	0.241	0.683
	Object Property Assertion	0.057	0.010	0.058	0.119
75% Random Noise	Membership	0.237	0.122	0.255	0.651
	Object Property Assertion	0.055	0.011	0.055	0.116
100% Random Noise	Membership	**0.226**	0.115	0.235	0.623
	Object Property Assertion	**0.054**	0.010	0.053	0.110
25% GNN Noise	Membership	0.244	0.126	0.254	0.732
	Object Property Assertion	0.066	0.014	0.070	0.146
50% GNN Noise	Membership	0.257	0.137	0.281	0.735
	Object Property Assertion	0.065	0.014	0.070	0.142
75% GNN Noise	Membership	0.258	0.134	0.294	0.742
	Object Property Assertion	0.065	0.014	0.069	0.139
100% GNN Noise	Membership	0.265	0.139	0.315	0.707
	Object Property Assertion	0.067	0.014	0.072	0.146
25% Logical Noise	Membership	0.285	0.149	0.369	0.758
	Object Property Assertion	0.071	0.017	0.078	0.156
50% Logical Noise	Membership	0.301	0.165	0.386	0.732
	Object Property Assertion	0.070	0.015	0.077	0.156
75% Logical Noise	Membership	0.315	0.170	0.421	0.745
	Object Property Assertion	0.072	0.016	0.080	0.161
100% Logical Noise	Membership	0.324	0.172	0.464	0.756
	Object Property Assertion	0.071	0.015	0.081	0.160

Table 10. Results on *pizza_250* ontology using *OWL2Vec** reasoner.

		MRR	**Hits@1**	**Hits@5**	**Hits@10**
No Noise	Membership	0.254	0.151	0.532	0.984
	Object Property Assertion	0.280	0.278	0.278	0.278
25% Random Noise	Membership	0.234	0.149	0.479	0.884
	Object Property Assertion	0.262	0.259	0.259	0.259
50% Random Noise	Membership	0.228	0.152	0.449	0.796
	Object Property Assertion	0.130	0.125	0.125	0.125
75% Random Noise	Membership	0.205	0.132	0.390	0.714
	Object Property Assertion	0.146	0.143	0.143	0.143
100% Random Noise	Membership	0.188	0.118	0.379	0.684
	Object Property Assertion	0.072	0.067	0.069	0.077
25% GNN Noise	Membership	0.216	0.125	0.454	0.847
	Object Property Assertion	0.474	0.462	0.513	0.537
50% GNN Noise	Membership	0.188	0.111	0.389	0.725
	Object Property Assertion	0.334	0.322	0.363	0.393
75% GNN Noise	Membership	0.170	0.102	0.344	0.648
	Object Property Assertion	0.206	0.193	0.223	0.252
100% GNN Noise	Membership	0.152	0.091	0.312	0.584
	Object Property Assertion	0.211	0.197	0.229	0.258
25% Logical Noise	Membership	0.110	0.078	0.206	0.333
	Object Property Assertion	0.012	0.010	0.010	0.013
50% Logical Noise	Membership	0.069	0.046	0.118	0.197
	Object Property Assertion	0.007	0.005	0.006	0.007
75% Logical Noise	Membership	0.051	0.031	0.089	0.141
	Object Property Assertion	0.004	0.003	0.004	0.004
100% Logical Noise	Membership	**0.040**	0.019	0.062	0.108
	Object Property Assertion	**0.003**	0.002	0.003	0.003

Table 11. Results on *pizza_250* ontology using *Box2EL* reasoner.

		MRR	Hits@1	Hits@5	Hits@10
No Noise	Membership	0.114	0.000	0.070	0.922
	Object Property Assertion	0.060	0.025	0.074	0.141
25% Random Noise	Membership	0.114	0.012	0.095	0.808
	Object Property Assertion	0.021	0.009	0.021	0.036
50% Random Noise	Membership	0.119	0.027	0.129	0.723
	Object Property Assertion	0.017	0.004	0.015	0.029
75% Random Noise	Membership	0.107	0.024	0.130	0.641
	Object Property Assertion	0.017	0.004	0.016	0.030
100% Random Noise	Membership	0.118	0.042	0.166	0.615
	Object Property Assertion	0.016	0.004	0.013	0.028
25% GNN Noise	Membership	0.098	0.000	0.058	0.766
	Object Property Assertion	0.060	0.025	0.076	0.146
50% GNN Noise	Membership	0.085	0.000	0.047	0.651
	Object Property Assertion	0.062	0.026	0.079	0.148
75% GNN Noise	Membership	0.076	0.000	0.040	0.574
	Object Property Assertion	0.062	0.027	0.079	0.147
100% GNN Noise	Membership	0.069	0.000	0.043	0.515
	Object Property Assertion	0.059	0.022	0.078	0.148
25% Logical Noise	Membership	0.055	0.004	0.054	0.283
	Object Property Assertion	0.001	0.000	0.000	0.000
50% Logical Noise	Membership	0.045	0.008	0.048	0.156
	Object Property Assertion	0.001	0.000	0.000	0.000
75% Logical Noise	Membership	0.039	0.007	0.039	0.107
	Object Property Assertion	0.001	0.000	0.000	0.000
100% Logical Noise	Membership	**0.032**	0.006	0.023	0.071
	Object Property Assertion	**0.001**	0.000	0.000	0.000

Table 12. Results on *pizza_250* ontology using *R-GCN*.

		MRR	**Hits@1**	**Hits@5**	**Hits@10**
No Noise	Membership	0.267	0.128	0.331	0.807
	Object Property Assertion	0.067	0.015	0.075	0.148
25% Random Noise	Membership	0.235	0.107	0.263	0.752
	Object Property Assertion	0.060	0.012	0.064	0.130
50% Random Noise	Membership	**0.220**	0.100	0.232	0.692
	Object Property Assertion	0.055	0.011	0.057	0.116
75% Random Noise	Membership	0.222	0.102	0.257	0.662
	Object Property Assertion	0.054	0.010	0.055	0.113
100% Random Noise	Membership	0.232	0.109	0.273	0.643
	Object Property Assertion	**0.053**	0.010	0.055	0.112
25% GNN Noise	Membership	0.258	0.121	0.324	0.783
	Object Property Assertion	0.065	0.013	0.072	0.146
50% GNN Noise	Membership	0.256	0.122	0.318	0.760
	Object Property Assertion	0.064	0.013	0.069	0.141
75% GNN Noise	Membership	0.266	0.126	0.349	0.761
	Object Property Assertion	0.064	0.014	0.068	0.142
100% GNN Noise	Membership	0.256	0.119	0.327	0.742
	Object Property Assertion	0.064	0.014	0.070	0.142
25% Logical Noise	Membership	0.273	0.128	0.381	0.778
	Object Property Assertion	0.069	0.015	0.077	0.155
50% Logical Noise	Membership	0.290	0.150	0.382	0.731
	Object Property Assertion	0.069	0.016	0.079	0.155
75% Logical Noise	Membership	0.303	0.150	0.435	0.726
	Object Property Assertion	0.068	0.015	0.077	0.154
100% Logical Noise	Membership	0.297	0.154	0.400	0.705
	Object Property Assertion	0.068	0.016	0.075	0.149

Table 13. Results on *OWL2Bench* ontology using *OWL2Vec** reasoner.

		MRR	Hits@1	Hits@5	Hits@10
No Noise	Membership	0.280	0.171	0.727	0.996
	Object Property Assertion	0.701	0.700	0.809	0.836
25% Random Noise	Membership	0.251	0.158	0.646	0.815
	Object Property Assertion	0.547	0.543	0.642	0.680
50% Random Noise	Membership	0.206	0.129	0.508	0.649
	Object Property Assertion	0.446	0.450	0.528	0.563
75% Random Noise	Membership	0.188	0.119	0.451	0.596
	Object Property Assertion	0.380	0.380	0.453	0.490
100% Random Noise	Membership	0.179	0.111	0.446	0.543
	Object Property Assertion	0.361	0.360	0.441	0.484
25% GNN Noise	Membership	0.288	0.167	0.774	0.943
	Object Property Assertion	0.510	0.512	0.581	0.605
50% GNN Noise	Membership	0.286	0.187	0.744	0.919
	Object Property Assertion	0.405	0.401	0.487	0.513
75% GNN Noise	Membership	0.248	0.138	0.696	0.890
	Object Property Assertion	0.344	0.342	0.391	0.407
100% GNN Noise	Membership	0.259	0.144	0.689	0.865
	Object Property Assertion	**0.276**	0.276	0.306	0.320
25% Logical Noise	Membership	0.211	0.112	0.518	0.761
	Object Property Assertion	0.637	0.648	0.779	0.833
50% Logical Noise	Membership	0.174	0.069	0.420	0.627
	Object Property Assertion	0.649	0.662	0.794	0.850
75% Logical Noise	Membership	0.161	0.060	0.351	0.609
	Object Property Assertion	0.651	0.658	0.799	0.856
100% Logical Noise	Membership	**0.154**	0.056	0.335	0.554
	Object Property Assertion	0.661	0.658	0.814	0.879

Table 14. Results on *OWL2Bench* ontology using *Box2EL* reasoner.

		MRR	**Hits@1**	**Hits@5**	**Hits@10**
No Noise	Membership	0.175	0.038	0.521	0.823
	Object Property Assertion	0.216	0.200	0.229	0.254
25% Random Noise	Membership	0.182	0.099	0.395	0.619
	Object Property Assertion	0.171	0.166	0.174	0.185
50% Random Noise	Membership	0.139	0.075	0.272	0.465
	Object Property Assertion	0.179	0.160	0.188	0.248
75% Random Noise	Membership	0.129	0.080	0.241	0.396
	Object Property Assertion	0.086	0.079	0.086	0.093
100% Random Noise	Membership	0.118	0.077	0.206	0.336
	Object Property Assertion	0.080	0.072	0.081	0.090
25% GNN Noise	Membership	0.182	0.064	0.486	0.785
	Object Property Assertion	0.148	0.136	0.149	0.166
50% GNN Noise	Membership	0.176	0.063	0.456	0.751
	Object Property Assertion	0.110	0.101	0.106	0.110
75% GNN Noise	Membership	0.169	0.058	0.436	0.713
	Object Property Assertion	0.086	0.078	0.081	0.089
100% GNN Noise	Membership	0.158	0.049	0.395	0.666
	Object Property Assertion	0.081	0.072	0.078	0.087
25% Logical Noise	Membership	0.120	0.037	0.276	0.481
	Object Property Assertion	0.012	0.003	0.020	0.026
50% Logical Noise	Membership	0.099	0.035	0.199	0.353
	Object Property Assertion	**0.002**	0.000	0.000	0.000
75% Logical Noise	Membership	0.086	0.033	0.150	0.280
	Object Property Assertion	**0.002**	0.000	0.000	0.000
100% Logical Noise	Membership	**0.061**	0.027	0.085	0.157
	Object Property Assertion	0.003	0.000	0.002	0.004

Table 15. Results on *OWL2Bench* ontology using *R-GCN*.

		MRR	Hits@1	Hits@5	Hits@10
No Noise	Membership	0.252	0.086	0.449	0.945
	Object Property Assertion	0.435	0.312	0.573	0.693
25% Random Noise	Membership	0.242	0.074	0.445	0.910
	Object Property Assertion	0.460	0.341	0.596	0.702
50% Random Noise	Membership	0.233	0.069	0.427	0.872
	Object Property Assertion	0.430	0.317	0.550	0.671
75% Random Noise	Membership	0.213	0.051	0.394	0.838
	Object Property Assertion	0.446	0.345	0.552	0.654
100% Random Noise	Membership	**0.206**	0.050	0.371	0.796
	Object Property Assertion	0.440	0.336	0.553	0.656
25% GNN Noise	Membership	0.256	0.088	0.458	0.943
	Object Property Assertion	0.443	0.307	0.595	0.709
50% GNN Noise	Membership	0.255	0.088	0.465	0.938
	Object Property Assertion	0.430	0.297	0.577	0.711
75% GNN Noise	Membership	0.261	0.091	0.482	0.941
	Object Property Assertion	0.453	0.324	0.605	0.721
100% GNN Noise	Membership	0.262	0.094	0.468	0.938
	Object Property Assertion	0.467	0.336	0.617	0.747
25% Logical Noise	Membership	0.239	0.074	0.429	0.909
	Object Property Assertion	0.293	0.182	0.397	0.546
50% Logical Noise	Membership	0.235	0.071	0.409	0.913
	Object Property Assertion	0.219	0.116	0.318	0.441
75% Logical Noise	Membership	0.240	0.076	0.420	0.910
	Object Property Assertion	0.186	0.098	0.260	0.393
100% Logical Noise	Membership	0.252	0.085	0.429	0.908
	Object Property Assertion	**0.150**	0.072	0.218	0.335

Figure 6. Variability of MRR on *family* ontology.

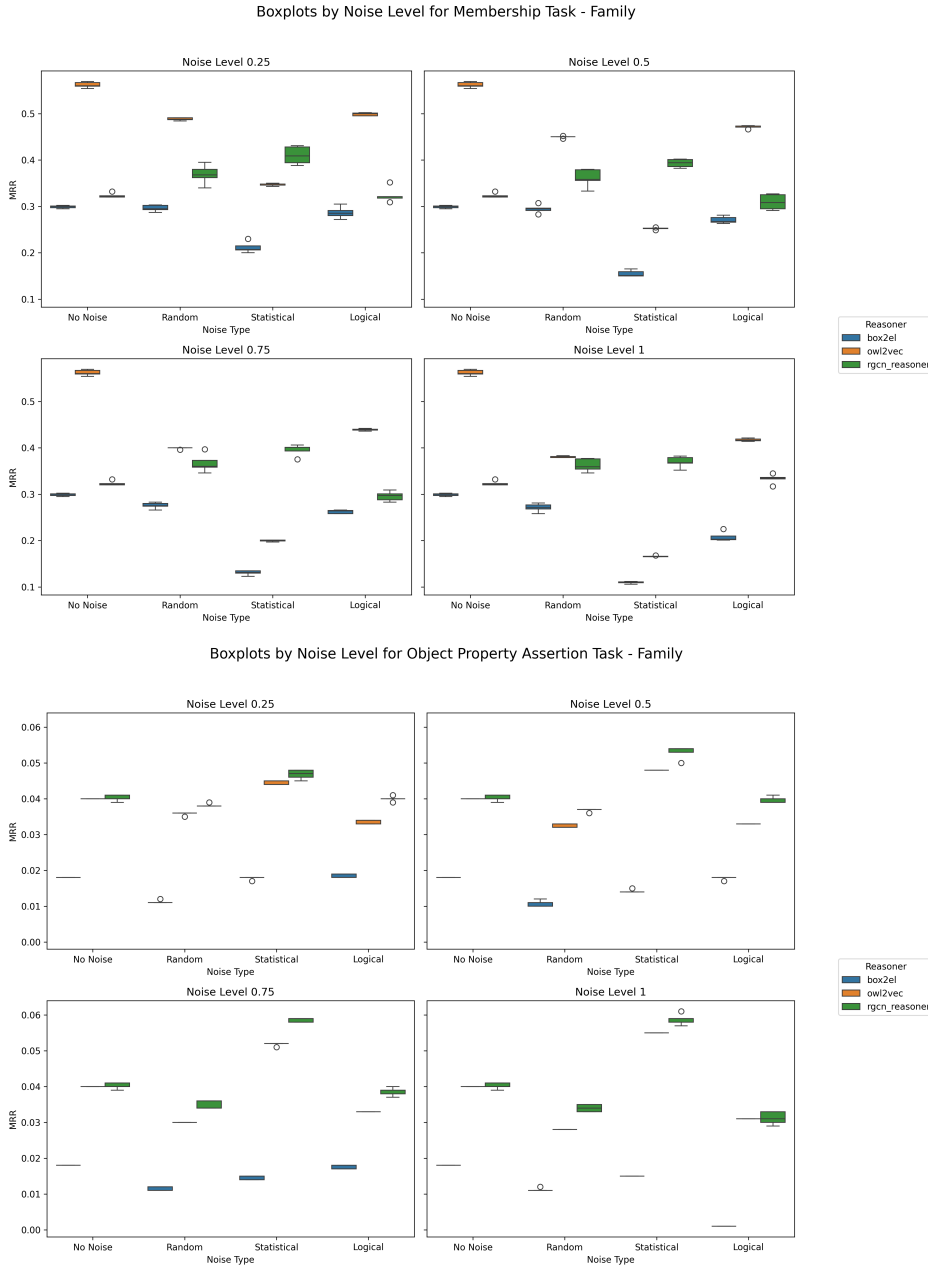

Figure 7. Variability of MRR on *pizza_100* ontology.

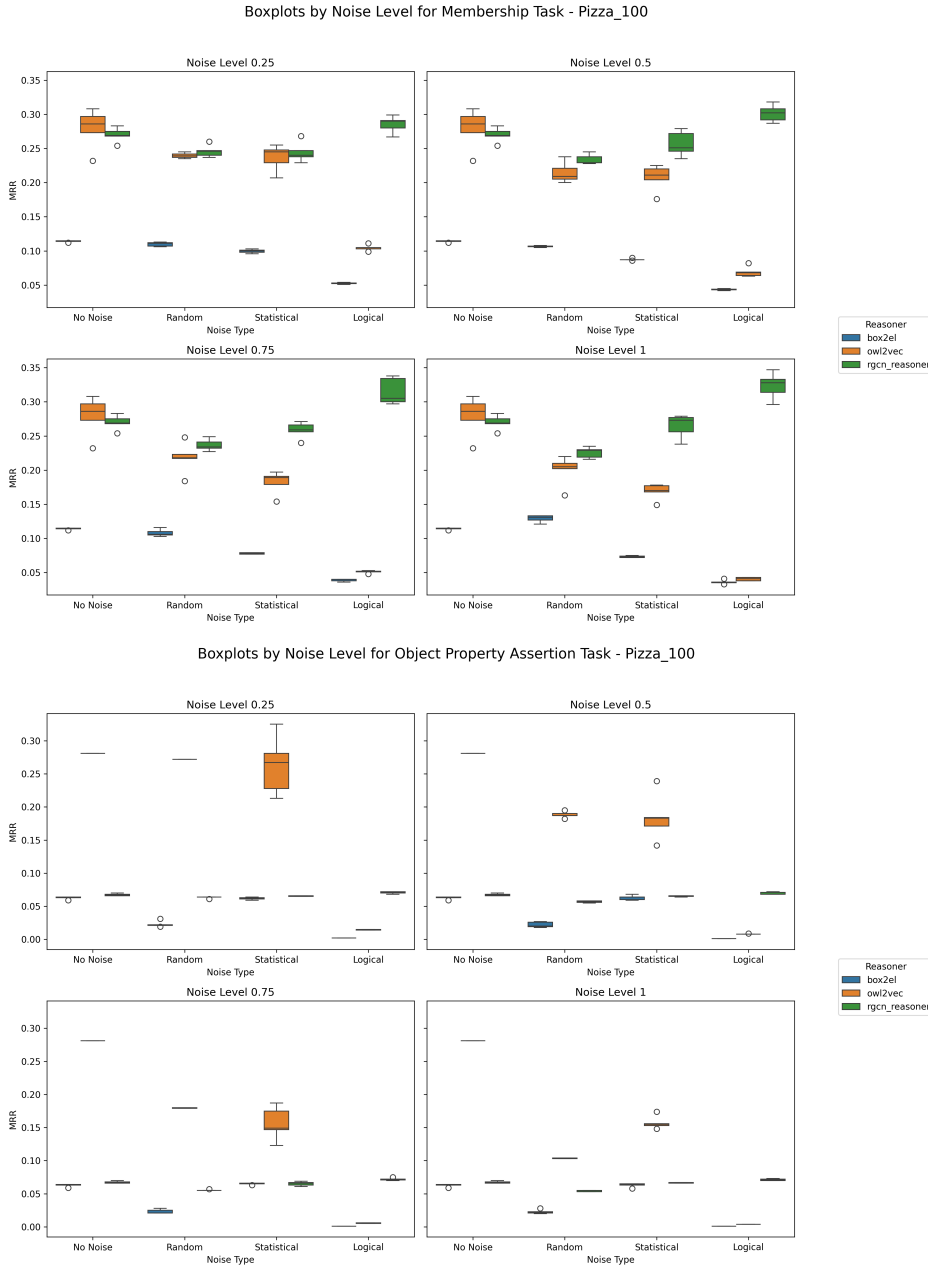

Figure 8. Variability of MRR on *pizza_250* ontology.

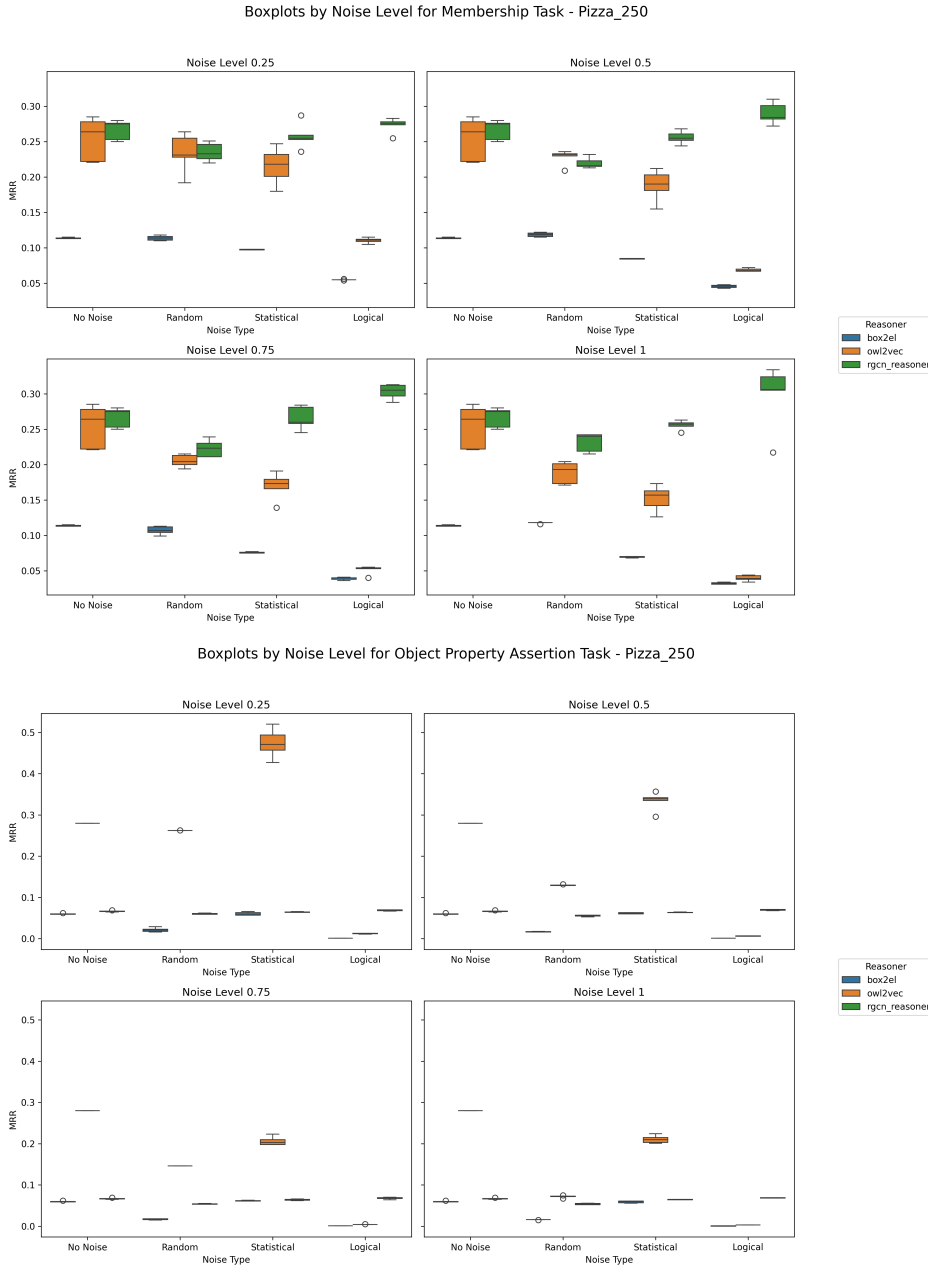

Figure 9. Variability of MRR on *OWL2Bench* ontology.