

## RESEARCH ARTICLE

# Comparative Analysis of Logic Reasoning and Graph Neural Networks for Ontology-Mediated Query Answering With a Covering Axiom

OLGA GERASIMOVA<sup>1</sup>, NIKITA SEVERIN<sup>1,2</sup>, AND ILYA MAKAROV<sup>2,3,4</sup>

<sup>1</sup>School of Data Analysis and Artificial Intelligence, HSE University, 101000 Moscow, Russia

<sup>2</sup>ISP RAS Research Center for Trusted Artificial Intelligence, 109004 Moscow, Russia

<sup>3</sup>Artificial Intelligence Research Institute, 105064 Moscow, Russia

<sup>4</sup>AI Center, NUST MISIS, 119049 Moscow, Russia

Corresponding author: Olga Gerasimova (ogerasimova@hse.ru)

This work was supported by a grant for research centers in the field of artificial intelligence, provided by the Analytical Center for the Government of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000D730321P5Q0002) and the agreement with the Ivannikov Institute for System Programming of the Russian Academy of Sciences dated November 2, 2021 No. 70-2021-00142.

**ABSTRACT** The problem of query answering over incomplete attributed graph data is a challenging field of database management systems and artificial intelligence. When there are rules on data structure expressed in the form of the ontology, the theoretical complexity of finding exact solution satisfying ontology constraints increases. Logic-based methods use theoretical constructions to obtain efficient rewritings of the original queries with respect to ontology and find an answer to the rewriting query over incomplete data. However, there is an opportunity to use faster machine learning methods to label all the data and query over the “most probable” data model without taking into account the ontology. This research paper investigates the effectiveness and trustworthiness of both mentioned approaches for answering ontology-mediated queries on graph databases that integrate an ontology with a covering axiom, which states that every node belongs to either of two classes. The first approach involves finding precise answers through logical reasoning and rewriting the problem into a datalog program, while the second approach employs a trained graph neural network to label data in a binary classification problem and leverages SQL for query answering. We conduct an in-depth analysis of the time performance of these approaches and evaluate the impact of training set selection on their ability of correct query answering. By comparing these approaches across various experiments, we provide insights into their strengths and limitations for answering ontology-mediated queries containing a Boolean conjunctive query. In particular, we showed the importance of logic-based approaches for ontology with a covering axiom and the inability of machine learning methods to find answers for ontology-mediated queries in large networks.

**INDEX TERMS** Computational complexity, datalog reasoner, disjunctive datalog, graph machine learning, graph neural networks, node classification, ontology-mediated query.

## I. INTRODUCTION

The object of our study lies in answering queries mediated by a Description Logic (DL) ontology in the framework of the Ontology-Based Data Access (OBDA) paradigm. Nowadays, OBDA is not just a theoretically interesting approach, but foremost a widely-applied advanced model that helps

The associate editor coordinating the review of this manuscript and approving it for publication was Giacomo Fiumara<sup>1</sup>.

in efficient and flexible data organisation and access, using ontology as a key advantage. That is why, OBDA works with not ordinary queries, but with Ontology-Mediated Queries (OMQs), each of which is a pair of a query and an ontology. Unfortunately, due to OBDA complexity, the benefits of this approach can be applied only for ontologies formulated in OWL 2 QL Web Ontology Language designed specifically for OBDA to preserve first-order rewriting of OMQs into equivalent standard queries.

In this paper, we focus on Boolean conjunctive queries mediated by a simple covering axiom stating that one class is covered by the union of two other classes. Chosen ontology rule is not covered by the functionality of OWL 2 QL, and we consider this case as an extension of the OBDA approach for more expressive and practical OMQs.

Theoretical research in this direction was started in [1], [2], and authors following a non-uniform approach presented results on data complexity and rewritability for concrete OMQs. Later, they showed in [3] that the data complexity of the problem varies from simple first-order rewriting in  $AC^0$  to a coNP case, in which there is no tractable solution. In this research, we are going to use tractable results from [3] and consider this case from another point of view.

We focus on OMQs belonging to the complexity classes L/NL and P, for which there exists a way to rewrite the original query to a datalog program taking into account information from disjunctive ontology. Having produced a datalog program, we aim to understand the efficiency of reasoning solvers for ontology-mediated query answering on different datasets and query patterns.

The ontological approach requires a great deal of effort to find rewritings of OMQ, if possible, and then to prove the tractability of these rewritings. Taking into account breakthroughs of machine learning techniques that successfully solve many problems of answering queries over data labelled by trained classification models, it was decided to apply innovative machine learning models for our task and make the comparison.

The main research gap is how to apply machine learning techniques to the ontology-mediated query answering task with a covering axiom in such a way that machine learning models can effectively help in finding correct answers to OMQ while being faster than their logic-based analogues. Despite having various works on query embedding for knowledge graphs [4], [5], [6], there are no works dealing with OMQ answering with a covering axiom using machine learning approaches. We interpret our ontology as a foundation for binary classification to label the data and then query data over labelled data. Such an approach requires much less time compared to logic-based methods.

We formulate the following research questions:

1. How the size of unlabelled data impacts the reasoning solvers performance and helps in finding the correct answer if we know all ground-truth labels?
2. Which graph properties help to achieve the best labelling performance for graph neural networks, which models perform the best?
3. Whether saturating graphs with labels obtained from the node classification models can be directly used to find the answer in the labelled data without the usage of the ontology with a covering axiom?

To answer these questions, we conducted a series of experiments on a small graph of collaborations in a classroom and three extensive networks with positive, neutral, and negative

assortativity impacting the performance of trained graph neural networks.

Our first task is to analyse the performance of several datalog reasoners depending on the different tractable queries with respect to smart/direct datalog query rewriting and the percentage of data masked for an ontology evaluation.

The second task is to provide extensive ablation study on the graph neural networks and choose the best hyper-parameters and models for labelling graphs with node classes having only a train subset of the whole graph.

Finally, our third task is to compare answers obtained by logical reasoners with the help of ontology and answers via querying data labelled by graph neural networks on large networks containing a large number of labelled data.

As a result, our research study establishes a connection between theoretical advancements in Ontology-Based Data Access regarding ontology-mediated queries with a covering axiom and the practical application of machine learning. We analyse interesting outcomes of machine learning techniques as an unconventional approach for theoretical ontological query answering problems and shed light on the exceptional interplay between logic and machine learning.

The paper has the following structure. After the introduction to the research idea and motivation in Section I, we overview studies on related topics in Section II. Then, we formulate the problem statement and describe the preparation of experiments to solve them in Section III. Section IV contains comparative results based on two approaches in terms of running time performance and correct answers evaluation for ontology-mediated query answering to analyse specific of the ontology with a covering axiom. Finally, we make a conclusions on formulated research questions and discuss limitations in Section V.

## II. RELATED WORK

In the following, we briefly provide background on OBDA and highlight notable recent achievements in this area that are pertinent to our study from the perspective of some similar aspects of problem statements. Then, we present initial researches, where tractable cases of ontological queries dealing with a covering axiom were found. Next, we consider different rule-based reasoners including systems with support of covering axiom in the ontology and their specifics of work. Finally, we describe modern graph neural network approaches that allow to label graph data performing node classification and then querying labelled data without the help of an ontology at all.

### A. ONTOLOGY-BASED DATA ACCESS

The fundamental direction that underlies our work is Ontology Based-Data Access [7], [8] providing answering queries mediated by a DL ontology [9]. The methodology of OBDA has being investigated in both theoretical and practical directions for applying OBDA to expressive ontologies beyond

standard OWL 2 QL<sup>1</sup> Web Ontology Language. The core idea of OBDA is the existence of a first-order rewriting or some datalog rewriting that provides tractability of answering ontology-mediated queries, that is why there are a lot of investigations on deciding a rewriting of OMQs over expressive ontologies or its construction if it is possible.

Toman and Weddell in their recent paper [10] on Horn ontologies have used Clark's completion of datalog programs and Beth's definability for deciding uniform FO-rewritability of OMQ in Horn-SHIQ and in Horn-DLFD and have provided an algorithm for rewritings construction based on their characterization of FO-rewritability. Also, they showed that their techniques can be applied to the non-uniform approach, in which we are interested during our research.

Authors of [11] gave us inspiration and ideas for our research because they have studied a similar problem, but with another original setting and have found a semantic characterisation of OMQs with an EL ontology for a complete classification in terms of the complexity and rewritability of ontology-mediated queries with atomic queries.

In the framework of computational complexity, it was provided new theoretical results [12] on decidability and complexity bounds for the entailment of positive existential two-way regular path queries (P2RPQs) mediated by expressive ontologies with transitive roles and qualified number restrictions that the problem is 2ExpTime-complete in combined complexity and coNP-complete in data complexity.

More tractable results on the extension of OBDA for an expressive ontology with a covering axiom were obtained in [3] and [13], where authors have presented results on rewritability and data complexity for specific ontology-mediated answering tasks to separate tractable and intractable cases for Boolean path queries. By the way, the current paper will continue to study the same case study, but from a practical point of view. We are going to use datalog rewritings obtained in [3] for conjunctive Boolean queries mediated with ontology containing a covering axiom and test them using datalog reasoning systems.

## B. DATALOG REASONERS

We aim to consider ontology rules and queries in terms of datalog syntax, hence, we overview available systems for datalog reasoning.

At the University of Oxford, researchers created a powerful tool for efficient processing and querying graph-structured data that is called RDFox [14]. This semantic reasoning engine boasts a unique patented in-memory architecture and parallelised computation and provides import Resource Description Framework (RDF) triples, rules, OWL 2, or Semantic Web Rule Language (SWRL) axioms using different formats including extensions of the datalog.

Protégé [15] is a ubiquitous open-source ontologies editor that has a lot of various plug-ins for knowledge representation and reasoning. For instance, Protégé's plug-in architecture

of SWRLTab [16] was designed to easily integrate various ontologies into the OMQ pipeline of data management. It allows us to take the output of Protégé with either rule systems or particular problem solvers and incorporate such framework in various applications using intelligent data access, verification, and saturation models. Ontop [17] plug-in is an application for classical OBDA with SPARQL queries over virtual RDF graphs. In addition, inside Protégé there are several OWL 2 reasoners supporting the rule-based approach such as hermit [18] and pellet [19].

Despite the many different rule-based tools providing datalog programming, in the framework of our research, we consider not trivial disjunctive datalog paradigm. In addition, we are interested in open source easy-to-use frameworks for querying large graph databases. Among such systems, we choose the following two.

A database system DLV (DataLog with  $\vee$ -disjunction) [20] has deductive reasoning using disjunctive logic programming. It is enterprise-level software supporting various ontology-based data access concepts and integrated with the NoSQL databases interface.

Another system was developed by researchers from the University of Potsdam as an Answer Set Programming (ASP) tool named Clingo [21]. It allows us to formulate a problem as a logic program. This tool enables users to convert logic programs with variables into equivalent propositional logic programs without variables. It then proceeds to compute the answer sets of the propositional programs.

In our research, both systems have very close input data formats and easy-to-use datalog syntax, thus providing useful tools for our experiments.

## C. GRAPH NEURAL NETWORKS

Graph Neural Networks (GNNs) have been successfully applied to various graph-based machine learning problems including node classification [22], which corresponds to our task. Graph Convolutional Network (GCN) [23], GraphSAGE [24] and Graph Attention Network (GAT) [25] are the most popular general-purpose GNNs.

Many graph machine learning methods construct node embeddings based on nodes local neighbourhoods. GCN stacks convolutional layers with shared node-wise feature transformation and operates with the full graph adjacency matrix at each of them. GraphSAGE, on the other hand, generalises neighbourhood aggregation by sampling only a subset of neighbouring nodes at different depth layers. GAT leverages self-attention layers over the node features to define the importance of neighbours. By concatenating the output of several different heads, this method captures different types of relationships between nodes.

Despite the wide use of the aforementioned models, it is often considered that standard GNNs only work well for homophilic graphs, i.e., graphs where nodes tend to be connected with the nodes of the same class. Recently, heterophilic graph learning has become an upward-trending research topic, and various specific structured GNNs have

<sup>1</sup><https://www.w3.org/TR/owl2-profiles/>

been proposed. Most of them either redefine the node neighbourhood (e.g. Geom-GCN [26]) or adapt messages between nodes (e.g. FAGCN [27]) to capture intra-class similarities [28]. Unlike them, several recent papers [28], [29] study inter-class node distinguishability and have found that inter-class edges can be helpful when the neighbour distribution satisfies certain conditions.

The authors of [29] revisited existing benchmarks and proposed an evaluation of a wide range of GNNs, both standard and heterophily-specific on the proposed benchmark. They revealed that general-purpose GNNs almost always outperform heterophily-specific methods on the proposed benchmark. Thus, we justify the choice of baseline models and provide our own results on the applicability of standard GNNs to the three large networks having different homophily/heterophily patterns.

### III. EXPERIMENT SETTINGS

We provide one small and three large networks for our experiments. Below, we discuss descriptive statistics of the data, state the core problem and experiment design for the comparative analysis, and describe quality assessment procedures.

#### A. DATASETS

##### 1) CLASSROOM<sup>2</sup>

A network describes connections between 26 classmates of 9 years of age at school and comes from a larger study of Dolata [30], where it was collected within questioning “With whom do you like to play with?”. The data is available in the form of an edges list and nodes have attributes in terms of gender.

The network was filtered to provide a toy example for the purposes of a balanced comparison of simple tractable queries over a small graph. In particular, we removed one direction of a symmetric edge (1042, 1006) (otherwise most OMQs will give the ‘yes’ answer). We mask nodes (remove original labels) from the layer interconnecting two communities of nodes with identical labels 1042, 1048, 1036, 1051, 1027, 1069, 1063, 1006. As a result, we have eight unlabelled nodes and different ‘yes’/‘no’ answers for simple OMQs. This masked dataset is called *Classroom* in our experiments, while Classroom Ground Truth (GT) stands for the original dataset after the edge removal and with all the labels known.

##### 2) POLITICAL BLOGS NETWORK<sup>3</sup>

The *Polblogs* network reflects directed interactions between US political weblogs recorded in 2005. There are two classes of node labels that correspond to the two political communities such as liberals and conservatives. Within the network, two blogs are connected with a directed edge, if the URL of the second blog is present on the page of the first blog. The collection process of Polblogs is described in the paper [31].

TABLE 1. Datasets statistics.

Statistics	Classroom	Polblogs	Deezer	Pokec-1
# nodes	25	1 224	28 281	265 388
# edges	87	16 718	185 504	700 352
Binary node labels	Gender [M, F]	Pol. parties [C, L]	Gender [0, 1]	Gender [M, F]
Labels ratio F/T	[0.48, 0.52]	[0.48, 0.52]	[0.56, 0.44]	[0.54, 0.46]
Assortativity	0.8623	0.8114	0.0304	-0.1795

##### 3) DEEZER EUROPE SOCIAL NETWORK<sup>4</sup>

The dataset presents a social network of music streaming app Deezer users. Available at SNAP project, it was collected using public API in March 2020. The *Deezer* dataset treats users from European countries as nodes and symmetric follower relationships between users as edges. Node features contain information on user preferences of artists presented in the app. Nodes were assigned by binary gender class using natural language processing methods based on the user name.

Used for node classification on graphs, this dataset is suitable for evaluation in our task due to possibly ambiguous labelling and hardness to correctly classify genders from graph information alone (with 20% train, it was reported only 65% best mean micro-averaged AUC [32]).

##### 4) POKEC SOCIAL NETWORK<sup>5</sup>

The dataset presents the most popular online social network in Slovakia. Pokec connects more than 1.6 million people over the past decade. A collection of data from the SNAP project contains anonymised data of the whole network, such as user gender, age, hobbies, interest, education, etc. All the relations between users are directed edges, which differ from the standard mutuality of friendship relations.

For our experiments, we retain the fourth part of this graph called *Pokec-1*, because, in the framework of our research, it is not necessary to maintain extra large networks.

Main statistics on the three graphs described above are represented in Table 1. For each binary class, we interpret them as either ‘F’ or ‘T’ labels as follows:

- *Classroom*: 0 - F (male), 1 - T (female)
- *Polblogs*: 0 - F (conservators), 1 - T (liberals)
- *Deezer*: 0 - F, 1 - T (authors did not specify which number 0 or 1 stands for males/females)
- *Pokec-1*: 0 - F (male), 1 - T (female)

### B. PROBLEM STATEMENT

The initial core fundamental research problem is finding tractable cases of OMQ answering with respect to data complexity and possible rewriting for the task of the form:

$$Q = (cov_A, q), \text{ where } cov_A = \{A \sqsubseteq F \sqcup T\} \text{ is an ontology and } q \text{ is a Boolean CQ with unary predicates } F, T \text{ and arbitrary binary predicates.}$$

Following the conjecture that interplays between the covering axiom  $A \sqsubseteq F \sqcup T$  and the structure of  $q$  determines

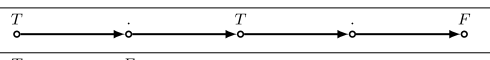

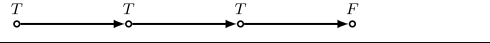
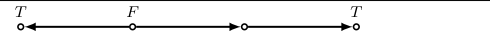
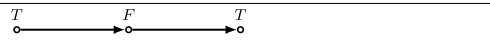
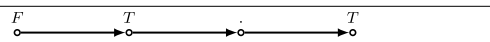
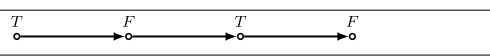
<sup>2</sup><https://statnet.org/workshop-intro-sna-tools/>

<sup>3</sup><http://konect.cc/networks/dimacs10-polblogs/>

<sup>4</sup><https://snap.stanford.edu/data/feather-deezer-social.html>

<sup>5</sup><https://snap.stanford.edu/data/soc-pokec.html>

**TABLE 2.** Query graphical representation as a directed labelled graph.

$q_0$		NL
$q_1$		NL
$q_2$		NL
$q_3$		NL
$q_4$		P
$q_5$		P
$q_6$		coNP

the complexity and rewritability properties of  $q$ , we have obtained the significant result on explicit AC<sup>0</sup>/NL/P/coNP-tetrachotomy of path-shaped queries (with disjoint  $F$  and  $T$ ) [3].

In our previous research [3], we highlighted the importance of tractable classes from the identified tetrachotomy and suggested datalog transformations for ontology-mediated queries with Boolean path conjunctive queries and a covering axiom. For the current experiments, we choose quite simple and short queries that can be processed by datalog systems in a reasonable time and that reflect different syntax patterns of query structure such as

- the ratio of the number of nodes with different labels that influences the complexity of answering task;
- specific placement of nodes labels that also can change significantly a datalog program structure and, hence, the complexity of answering task;
- empty nodes presence that makes query answering process more complex;
- edge directions that we can vary and, for instance, form linear path queries with tractable complexity.

Table 2 provides the complexity classes and graphical representations of conjunctive queries, for which we are carrying out experiments on performance evaluation of datalog reasoners in this study.

In order to perform experiments with machine learning approaches, we need to train graph neural networks on the train subgraph of the original graph and extrapolate node class predictions to the rest of the graph. After that, we can just run an SQL query over the labelled graph without the use of the ontology with a covering axiom.

### C. EXPERIMENT DESIGN

For the ontology approaches, logic-based solvers can find an answer for any size of labelled data; however, the size of unlabelled data can affect the search time. Taking that into account, we propose the idea of using GNN-based methods to fill the missing labels. We substitute the task of querying via OMQ rewriting with training machine learning models for the binary node classification and directly querying over labelled data.

In our research, one of the ideas was to analyse the influence of the size of unlabelled data on the results of approaches and to find out the possible ‘balance’ between logic-based and GNN-based methods’ performances.

Basically, when almost no labels are known, machine learning will fail, similar to logic solvers. When almost 100% labels are known, machine learning does not make any difference compared to querying over original data. So, we aimed to study whether there is an interesting threshold of training set size and masking parameters benefiting machine learning approaches which, by the nature of constructing, are much faster than logic-based solvers.

We consider settings where we mask 5% to 95% of graph data with step 5%, taking *three* random seeds. In addition, during masking, we need to consider the *ratio of masked binary classes*. In the original datasets, this ratio is around 0.5; thus, we consider a balanced masking procedure with a ratio of 0.5 and two imbalanced ratios of 0.25 and 0.75, respectively. Finally, we study how the assortativity coefficient impacts the GNN models performance and provides GNNs designed for homophilic and heterophilic networks.

With a fixed train set, we train GNN models, label the whole graph and compare the time complexity and correctness of OMQ via reasoner on a train set with the original data labelling and with the result of querying labelled by GNN graph.

### D. ASSESSMENT PROCEDURES

In order to conduct the comparative study, evaluating the quality of both ontology-based reasoner and labelling by GNNs, we propose the following three aspects that should be studied.

The first two concepts evaluate the precision of each method, while the third one aims to estimate how well the GNN-based reasoning approximates the logic-based one:

- consistency of predictions of ontology-based reasoner compared to ground truth labels;
- consistency of predictions of GNN-based reasoner compared to ground truth;
- consistency of predictions of GNN-based reasoner with ontology-based reasoner’s predictions as ground truth.

To evaluate the quality of GNN models for the node classification task, we have used a standard balanced  $F_1$  measure. We selected default parameters and most popular settings taken from surveys on graph machine learning [22], [33], [34], [35].

In addition, measuring the running time for answering OMQs is one of the important metrics for analysing the reasoners performance, because the tractability of OMQs relates to the time of computations. In the case of GNN models, we do not take into account time for training and labelling over masked data; we consider only time for reasoning over already labelled data.

#### IV. EXPERIMENT RESULTS

First, we evaluate the performance of reasoners on large networks and when their answers on masked data are inconsistent with labelled data.

Second, we perform an ablation study to choose the best GNN models based on node classification tasks. We provide results on the limitations of the applicability of GNN-based labelling on large networks.

Finally, we conduct the experiment with a small graph to compare all the settings with various ‘yes’/‘no’ answers to OMQs, including Boolean conjunctive queries presented in Table 2. We make conclusions on the consistency of GNN-based and logic-based approaches for OMQs answering with regard to ontology with a covering axiom.

##### A. PERFORMANCE ANALYSIS OF DATALOG REASONERS

Here, we analyse how the size of unlabelled data impacts the running time for the OMQs answering using logical reasoning solvers. In particular, we study how the masking percentage of graph data reflects on finding the correct answer with respect to ground truth labels.

In Table 3 and 4, we provide detailed time comparison for finding an answer for OMQs for datasets *Polblogs* and *Deezer*, respectively. We masked 25%, 50% and 75% of graph data with class ratios of 0.25, 0.50, and 0.75 for the masked data part. We report mean and standard deviation (std) computed over three random seeds for each setting.

The results for large networks are provided with respect to  $q_0$ – $q_5$  queries from Table 2 omitting  $q_6$  because it lies in coNP without the existence of tractable rewriting and runs out of device memory even for reasoning on 5% of masked data.

As one can see, the impact of the class ratio is unstable, and it is hard to make a universal assumption on how the class ratio impacts the running time for answering OMQs. Across all three datasets, we observe deviations from 10% to 50% of the time for a balanced ratio equaled 0.5 (consistent with all three datasets’ class ratios). It can be explained that the answering time for a particular query depends on the number and positions of labelled T or F literals in the query. Thus, it is obvious that changing the class ratio directly impacts the time for answering OMQ.

In addition, in complex query  $q_5$ , we observe a significant increase of std for the class ratio equaled 0.25 and 75% masked data compared to the balanced class ratio of 0.5 because the corresponding datalog rewriting takes too much time to find the answer in highly imbalanced unlabelled data.

In almost all cases, we can see a tendency that Clingo is significantly faster than DLV. We suppose that Clingo has improved performance optimizations and more efficient implementation of the grounding and solving steps in the two-step ASP process, resulting in quicker computation of stable models for our logic programs. All the answers between the two reasoners are consistent, which was expected from the logic approach.

Next, in Figure 1 and 2, we compare the speed of answering to different OMQs in dependence on the masking percentage for both reasoners. For simple queries  $q_1$ – $q_3$ , it does not significantly impact the answering time on *Polblogs* and *Deezer*. However, there is a decreasing time trend when the number of unlabelled data increases. It may be due to the fact, that the reachability problem underlying NL complexity of  $q_1$ – $q_3$  can be computed faster when less number of nodes are labelled.

For P-complexity queries  $q_4$  and  $q_5$ , we observe the increase of time, especially for  $q_5$ , because the underlying datalog rewriting spans a polynomial tree to find an answer to OMQ and the size of that tree increases when the number of unlabelled data increases.

Also, it is worth to describe the situation for  $q_0$ , when the running time is extra high in comparison to the other considered queries. It seems that it is due to the number of nodes in the query.  $q_0$  is the longest query containing only 5 nodes in our list, but we can see that just adding one node leads to a crucial timing increase compared to  $q_2$  and  $q_3$ . Besides, for directed *Polblogs* the computed time is much higher than for *Deezer* with symmetric edges.

In addition, we want to explain two anomalies in the aforementioned Figures.

One is a sudden increase of time near 25% mask ratio for  $q_2$  query on *Polblogs* with Clingo (see Figure 1 (left)). The reason behind this may be the fact that reachability from “ $T \rightarrow T \rightarrow T$ ” to ‘F’ requires a search for the directed path of three ‘T’, which is implemented faster in Clingo ASP compared to DLV ASP. It only holds for the case when many data are labelled, but starting from a certain increase of unlabelled data, its performance decreases and both reasoners perform similarly to each other. It is logical that increasing uncertainty in data leads to decreasing the number of patterns “ $T \rightarrow T \rightarrow T$ ” in data and, hence, decreasing the number of reasoning steps.

Another case is a sudden decrease of answering time after masking 85% of the *Deezer* graph for  $q_4$  and  $q_5$  (see Figure 2 (right) and 2 (left)). In such cases, the answer changes from ‘yes’ due to a lack of labels either T or F, thus making the answer ‘no’ due to insufficient labelled data (as shown later in Figure 3).

Basically, there is a threshold for masking, starting from which the reasoners answers will deviate from the answers on ground truth fully-labelled real-world graph. That may happen not only if either F or T is missing in the labelled data, but also with just several F/T labels presented.

One of the important conclusions of this section is that one needs to be sure that for a given query type there is enough labelled data for the usage of the logical approach. For social networks, usually, only 5-10% of people mention their gender in profile info; thus, direct application of the logical approach may provide incorrect answers. One of the possible ways to mitigate this problem may be the additional labelling via text mining for names, GNNs, and other methods helping to label more data.

TABLE 3. Running times of DLV and Clingo on Polblogs.

Query	M / F Ratio	DLV, Masked			Clingo, Masked		
		25%	50%	75%	25%	50%	75%
$q_0$	0.25	2385.704±69.704	2046.157±107.064	1252.533±82.160	592.466±24.114	501.292±15.901	323.944±18.517
	0.50	2837.41±59.413	2809.734±87.281	2318.423±94.859	633.047±5.843	672.103±20.843	598.614±46.230
	0.75	3370.457±183.876	3546.030±42.558	1064.141±106.536	648.101±8.020	754.136±5.824	285.793±33.640
$q_1$	0.25	0.443±0.003	0.466±0.002	0.642±0.085	0.388±0.015	0.374±0.011	0.391±0.013
	0.50	0.460±0.017	0.465±0.003	0.365±0.006	0.391±0.010	0.364±0.004	0.390±0.003
	0.75	0.464±0.011	0.338±0.014	0.593±0.014	0.417±0.036	0.41±0.037	0.411±0.007
$q_2$	0.25	37.222±3.965	17.467±2.408	5.746±0.609	8.236±1.903	2.145±0.214	0.723±0.066
	0.50	47.079±1.468	36.401±1.868	16.411±0.657	10.531±4.050	5.408±0.356	1.896±0.069
	0.75	56.500±0.933	54.092±1.109	4.451±0.769	9.441±0.176	9.478±0.553	0.598±0.059
$q_3$	0.25	2.293±0.059	2.708±0.395	1.861±0.045	0.395±0.005	0.411±0.004	0.434±0.005
	0.50	2.426±0.072	2.903±0.0420	2.684±0.021	0.397±0.004	0.411±0.018	0.444±0.017
	0.75	2.707±0.019	3.387±0.041	1.739±0.072	0.400±0.001	0.422±0.007	0.440±0.005
$q_4$	0.25	1.264±0.054	2.583±0.172	4.931±0.826	0.415±0.004	0.461±0.009	0.444±0.020
	0.50	1.224±0.169	2.773±0.070	6.224±0.160	0.410±0.006	0.447±0.012	0.478±0.013
	0.75	1.145±0.029	3.317±0.055	5.618±0.168	0.401±0.003	0.455±0.008	0.435±0.020
$q_5$	0.25	64.353±2.417	135.171±3.553	291.627±43.041	12.161±1.058	17.262±0.320	33.457±0.471
	0.50	54.600±5.499	150.945±3.568	399.764±13.664	12.078±0.381	24.546±0.302	39.197±0.630
	0.75	52.953±1.528	202.838±3.598	360.568±12.068	14.700±0.792	33.961±0.600	35.101±0.544

TABLE 4. Running times of DLV and Clingo on Deezer.

Query	O / I Ratio	DLV, Masked			Clingo, Masked		
		25%	50%	75%	25%	50%	75%
$q_0$	0.25	159.643±0.647	134.672±1.174	5.621±6.474	36.724±0.459	34.678±0.209	20.720±1.614
	0.50	205.028±3.326	257.879±6.469	281.83±6.189	47.095±0.758	65.447±1.511	72.865±1.395
	0.75	256.502±2.643	403.995±6.472	52.628±6.798	57.68±0.572	101.312±0.395	14.847±1.749
$q_1$	0.25	2.150±0.062	2.523±0.153	2.912±0.197	1.614±0.141	1.509±0.117	1.495±0.160
	0.50	2.172±0.083	2.718±0.062	1.177±0.040	1.459±0.118	1.636±0.134	1.73±0.069
	0.75	2.229±0.112	2.996±0.067	3.261±0.123	1.534±0.133	1.555±0.103	1.666±0.170
$q_2$	0.25	7.374±0.611	6.775±0.578	5.798±0.153	2.800±0.023	2.584±0.092	2.127±0.017
	0.50	9.267±0.707	9.500±0.282	9.411±0.252	2.783±0.033	2.909±0.112	2.230±0.076
	0.75	11.418±0.317	14.234±0.282	6.024±0.219	3.609±0.293	3.486±0.250	1.883±0.131
$q_3$	0.25	5.430±0.085	5.77±0.225	5.796±0.455	1.732±0.031	1.800±0.008	2.043±0.137
	0.50	5.839±0.074	6.647±0.02	6.756±0.057	1.585±0.119	1.724±0.114	2.000±0.105
	0.75	6.313±0.086	7.774±0.537	5.102±0.219	1.749±0.016	1.947±0.122	1.916±0.116
$q_4$	0.25	4.405±0.406	9.557±1.007	16.341±0.441	1.594±0.127	1.595±0.019	1.678±0.010
	0.50	4.565±0.230	8.614±1.145	15.298±0.428	1.531±0.077	1.667±0.035	2.030±0.120
	0.75	4.885±0.122	10.989±0.177	20.146±0.997	1.492±0.005	1.705±0.010	1.703±0.018
$q_5$	0.25	35.331±1.755	77.569±6.294	194.958±25.661	6.276±0.644	8.216±0.297	11.564±0.263
	0.50	27.027±1.198	87.333±10.967	185.997±0.975	6.980±0.411	12.974±0.472	16.013±0.116
	0.75	32.138±1.485	112.441±10.800	240.039±10.154	8.286±0.132	14.694±1.198	12.048±0.124

## B. TRAINING GRAPH NEURAL NETWORKS FOR DATA LABELING

Next, we decided to train GNNs to label masked node classes for our graphs. We chose several representative neural architectures as our baselines: GCN, GraphSAGE, GAT and two types of their modifications, proposed in [36].

The first kind of modification is to add a residual connection between neural network layers to additionally propagate node features as it was realised in ResNet [37]. The second modification is based on [38], which shows that separating ego- and neighbour-embeddings in the GNN aggregation step is beneficial when learning under heterophily. In our experiments, we added this modification to GCN and GAT in the same way as it was done in GraphSAGE, where a node embedding is concatenated to the mean of its neighbours embeddings, rather than simply summing them. We will refer

to these modifications by postfixes “-Skip” and “-SEP”, respectively.

In order to conduct a fair comparison, node features were initialised in graph neural networks based solely on the graph structure, as OMQs do not utilise node features for inference. Three types of initialisation were taken into account:

- random initialisation: node features are uniformly sampled from  $[0; 1]$ ;
- dummy initialisation: node features are represented as one-hot vectors with  $|V|$  components, where  $|V|$  is a number of nodes in a graph;
- node2vec-based initialisation: node features are obtained by training node2Vec model [39] maximising the probability of simultaneously finding neighbouring nodes to have appeared in the same random walks. Thus, such features preserve structural similarities between nodes.

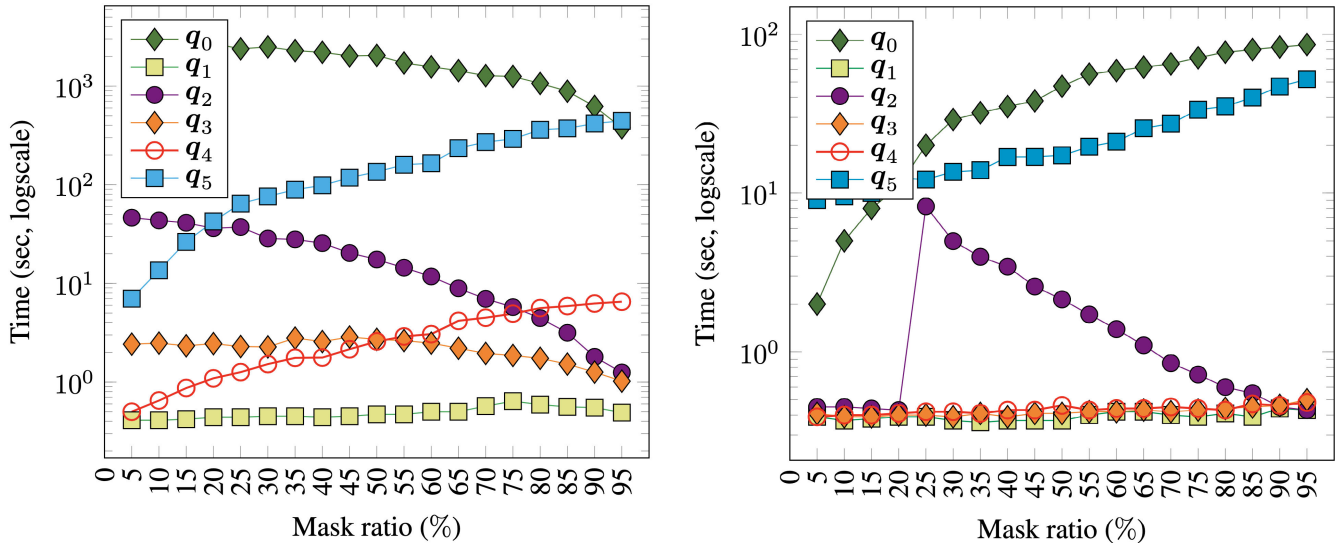


FIGURE 1. Comparison of the running time of DLV (left) and Clingo (right) systems for different queries on *Polblogs*.

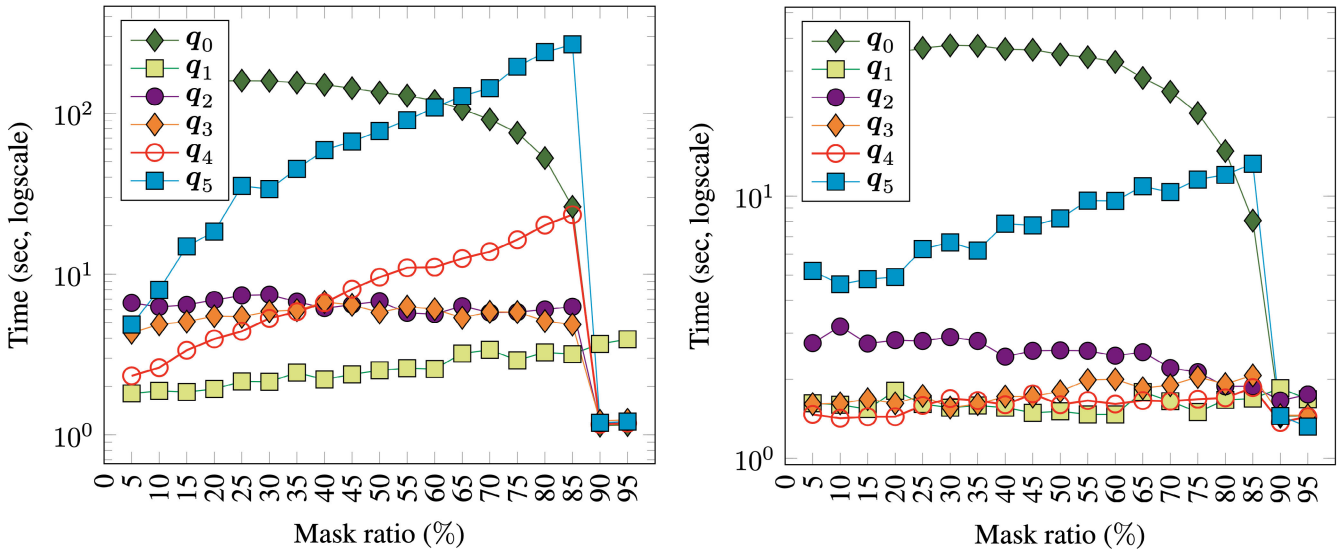


FIGURE 2. Comparison of the running time of DLV (left) and Clingo (right) systems for different queries on *Deezer*.

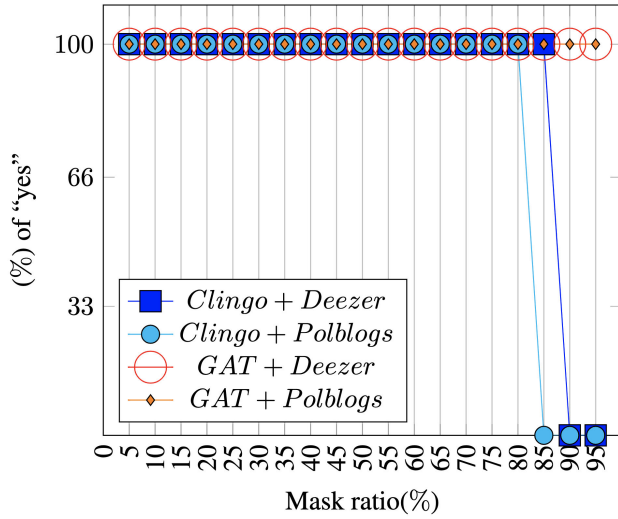
Prior to the comparison of GNN-based with logic-based one, for each architecture, we found a set of optimal parameters via ablation study presented in Table 5.

It can be clearly seen that different kinds of masking strongly affect the performance of all models on *Polblogs*, except for GAT and its modifications. Specifically, GAT and GAT-Skip show the best results, while the quality of GCN-based and GraphSAGE-based models drastically falls, as the masking ratio increases. This could be explained by the fact that the dataset has a high level of assortativity. In such cases, a more complex graph neural model is able to accurately capture this characteristic even when a significant amount of masking is applied.

On the *Deezer* dataset with assortativity close to 0, all GNNs show performance not higher than 0.57 with a small variance for different masking. It might be due to the irregular distribution of node labels, which challenges message passing to capture the underlying patterns and relationships from the local node neighbourhood. The best results are shown by vanilla GCN and GAT. The low performance of GraphSAGE-based models might indicate that the sampling node neighbourhood does not let them preserve the necessary structural patterns of this graph.

On *Pokec-1*, which has negative assortativity, the models show almost the same results, but worse metrics, thus we decided not to consider labelling it for further results because





**FIGURE 3.** Comparison of correct answers for logic reasoner Clingo and reasoning using labelled via GAT data. We see that on both datasets logic reasoning fails when the number of labelled data is small. GNN labelling just provides the answer ‘yes’ in all the cases due to generating all simple query patterns by the labelling procedure.

the node classification task is not trustworthy as a backbone for ontological query answering.

Thus, we have selected GCN, GAT, and GAT-Skip as the most promising models for performance-related experiments. Among these models, we mostly use GAT as it has shown to be the best performing (see Table 5).

The time complexity for training and inference of the best-performing models is presented in Table 6. In the case of GAT-based models, it is evident that the training time increases as the number of labelled nodes in the graphs increases. This is because of the general property of message passing that larger graphs take more time to be processed. On the other hand, for GCN, having either too small labelled data also leads to increased training time. This could be explained by the fact that GCN has simpler architecture than GAT, requiring more epochs for convergence when training on a small graph. At the same time, inference time is comparable between models trained on the training sets of various sizes.

As a result, each GNN model has made the labelling of given datasets and we achieved the main goal to query fully labelled networks. However, due to the complex and dense enough network structures that easily cover all syntax patterns regardless of concrete node labelling, all considered queries yielded a ‘yes’ response for every GNN model.

One can see that node classification did not work well for the precise answering of simple Boolean conjunctive queries combined with a covering axiom. Finding an accurate answer requires consideration of contextual dependencies and interactions between nodes, in particular nodes labels. Even state-of-the-art node classification methods can overlook or have difficulty capturing these dependencies, leading to inaccurate results. Moreover, sometimes a few wrong label predictions

**TABLE 5.** Comparison of various GNNs configurations for three large network datasets (the best results are in bold, the second best results are underlined).

Data	Model	Hidden Layers	Input Weights	Features	Normalization	Masking	F1-score	
Polblogs	GCN	[64]	false	dummy	LayerNorm	0.1 0.5 0.9	0.940 0.900 0.690	
	GCN+Skip	[64]	false	dummy	LayerNorm	0.1 0.5 0.9	<u>0.947</u> 0.933 0.767	
	GCN-SEP	[64, 64]	false	dummy	No	0.1 0.5 0.9	0.917 0.883 0.720	
	GCN-SEP+Skip	[64]	false	dummy	LayerNorm	0.1 0.5 0.9	<u>0.947</u> <u>0.937</u> 0.667	
	GraphSAGE	[64, 64]	false	dummy	No	0.1 0.5 0.9	0.940 0.893 0.747	
	GraphSAGE+Skip	[64, 64]	false	dummy	No	0.1 0.5 0.9	0.1 <u>0.950</u> 0.913 0.750	
	GAT	[64]	false	dummy	LayerNorm	0.1 0.5 0.9	<b>0.950</b> 0.926 0.923	
	GAT+Skip	[64]	false	dummy	LayerNorm	0.1 0.5 0.9	0.1 0.943 <b>0.940</b> <b>0.940</b>	
	GAT-SEP	[64, 64]	false	dummy	LayerNorm	0.1 0.5 0.9	0.1 0.940 0.933 0.870	
	GAT-SEP+Skip	[64]	false	dummy	BatchNorm	0.1 0.5 0.9	0.1 0.933 0.853 0.843	
	Deezer	GCN	[128, 128]	true	Node2Vec	BatchNorm	0.1 0.5 0.9	0.1 <b>0.561</b> 0.551
		GCN+Skip	[128]	false	Node2Vec	BatchNorm	0.1 0.5 0.9	0.1 0.566 0.560 0.543
GCN-SEP		[128]	true	Node2Vec	BatchNorm	0.1 0.5 0.9	0.1 0.567 0.555 0.537	
GCN-SEP+Skip		[128, 128]	true	Node2Vec	BatchNorm	0.1 0.5 0.9	0.1 0.566 0.559 0.532	
GraphSAGE		[128]	true	Node2Vec	BatchNorm	0.1 0.5 0.9	0.1 0.562 0.558 0.540	
GraphSAGE+Skip		[128, 128]	false	Node2Vec	BatchNorm	0.1 0.5 0.9	0.1 0.567 0.556 0.541	
GAT		[128]	true	dummy	BatchNorm	0.1 0.5 0.9	0.1 <b>0.570</b> 0.557 0.554	
GAT+Skip		[128]	true	dummy	BatchNorm	0.1 0.5 0.9	0.1 0.563 0.558 <b>0.557</b>	
GAT-SEP		[64]	true	dummy	BatchNorm	0.1 0.5 0.9	0.1 0.562 0.557 0.554	
GAT-SEP+Skip		[64, 64]	true	dummy	BatchNorm	0.1 0.5 0.9	0.1 0.566 0.557 <b>0.557</b>	
Polccc-1		GCN	[128, 128]	true	dummy	No	0.1 0.5 0.9	0.1 0.541 <b>0.542</b> 0.541
		GCN+Skip	[128, 128]	true	dummy	No	0.1 0.5 0.9	0.1 0.540 <b>0.542</b> 0.542
	GCN-SEP	[128, 128]	true	dummy	No	0.1 0.5 0.9	0.1 <b>0.541</b> 0.542 <b>0.542</b>	
	GCN-SEP+Skip	[128, 128]	true	dummy	No	0.1 0.5 0.9	0.1 <b>0.541</b> 0.542 <b>0.542</b>	
	GraphSAGE	[128]	true	dummy	No	0.1 0.5 0.9	0.1 0.540 0.539 0.542	
	GraphSAGE+Skip	[128, 128]	true	random	No	0.1 0.5 0.9	0.1 <b>0.541</b> 0.542 0.530	
	GAT	[128, 128]	true	dummy	No	0.1 0.5 0.9	0.1 <b>0.541</b> 0.542 0.541	
	GAT+Skip	[128, 128]	true	random	No	0.1 0.5 0.9	0.1 <b>0.541</b> 0.542 0.541	
	GAT-SEP	[128, 128]	true	dummy	No	0.1 0.5 0.9	0.1 <b>0.541</b> 0.542 0.542	
	GAT-SEP+Skip	[128, 128]	true	random	No	0.1 0.5 0.9	0.1 <b>0.541</b> 0.542 0.536	

are enough to get an incorrect answer to a query, even for models with almost 100% accuracy.

This is not an obvious fact that became one of the findings of our work. It appears that having disjunctive ontology property, evaluation of the Boolean query checks the data property for all the possible labelling cases, while the GNN-based approach checks the “most probable” data model.

The GNN-based approach can still be applied to either small graphs with transparent structure or significantly more complicated conjunctive queries, but tractable cases were identified among only path- or tree-shaped conjunctive

**TABLE 6.** Speed of GNNs (seconds) with different training set sizes on three network datasets.

Dataset	Mask	Phase	GCN	GAT	GAT-Skip
Polblogs	10%	Train	1.596	1.823	1.792
		Inference	0.008	0.001	0.007
	50%	Train	0.976	1.471	1.502
		Inference	0.009	0.016	0.006
	90%	Train	1.053	1.548	1.646
		Inference	0.009	0.009	0.008
Deezer	10%	Train	2.115	2.215	2.203
		Inference	0.009	0.009	0.008
	50%	Train	1.175	1.353	1.331
		Inference	0.012	0.011	0.010
	90%	Train	1.198	1.215	1.224
		Inference	0.016	0.016	0.013
Pokey-1	10%	Train	16.694	25.930	18.809
		Inference	0.103	0.078	0.213
	50%	Train	13.265	15.048	9.415
		Inference	0.229	0.266	0.123
	90%	Train	29.142	12.570	2.332
		Inference	0.130	0.242	0.190

queries, for which we could test their datalog rewritings. To evaluate the performance of GNNs on the small graph, we perform rigorous evaluation in the next section.

### C. COMPARISON OF LOGIC-BASED AND GNN-BASED QUERY ANSWERING WITH A COVERING AXIOM ON A SMALL GRAPH CLASSROOM

First, we take the small graph *Classroom* and compare two reasoners DLV and Clingo for two versions of datalog representations of OMQs: one of which is the *original* combination of the query and the covering axiom rule, and another is *rewriting*. Let us explain in detail.

The idea of the original datalog program is to explicitly use covering axiom as a disjunctive datalog rule, thus generating  $2^{|A|}$  data models, where  $|A|$  is a number of unlabelled data. Another direction is to use tractable datalog rewritings suggested in [3], which is the efficient way that will be used in further experiments. The time comparison is presented in Table 7. As one can see, Clingo is a little bit faster than DLV (which is more noticeable in Section IV-A).

For  $q_6$  there is no known polynomial rewriting due to coNP complexity, thus only the original datalog program is possible, but it is still tractable for very small graphs like *Classroom*. These experiments are only possible on small graph data because for larger graphs the computations would be intractable. For e.g., for 5% of masked *Polblogs* data, direct answering of OMQ via the original datalog program, which checks all data models arising from the covering axiom, takes over 3 hours spending 30Gbs in intermediate computations (while rewritten OMQ takes just seconds).

Second, we trained several GNNs that were shown to perform the best on large networks (see Section IV-B) and compare the consistency of the answers between reasoner-based (finding answer to OMQ with missing labels in graph and the ontology at hand), and GNN-based (finding answer to query over labelled by GNN graph).

**TABLE 7.** Comparison of DLV and Clingo, both with *original* and rewritten datalog programs for OMQs with a covering axiom.

Query	DLV, Original	DLV, Rewriting	Clingo, Original	Clingo, Rewriting
$q_0$	0.17	0.14	0.15	0.15
$q_1$	0.15	0.14	0.15	0.15
$q_2$	0.15	0.15	0.15	0.14
$q_3$	0.16	0.15	0.15	0.15
$q_4$	0.15	0.14	0.15	0.14
$q_5$	0.15	0.14	0.15	0.15
$q_6$	0.16	no rewriting	0.15	no rewriting

**TABLE 8.** Comparison of querying *Classroom* with DLV and Clingo reasoners (with rewritten datalog) with three GNN models. Column 'Data' represents querying data with eight unlabelled nodes without ontology. The 'GT' column (Ground Truth) represents answers to queries over the original fully-labelled graph.

Query	DLV	Clingo	GCN	GAT	GAT skip	Data	GT
$q_0$	yes	yes	yes	yes	yes	no	yes
$q_1$	yes	yes	yes	yes	yes	no	yes
$q_2$	yes	yes	yes	yes	yes	no	yes
$q_3$	no	no	yes	yes	yes	no	yes
$q_4$	no	no	yes	yes	yes	no	no
$q_5$	no	no	yes	yes	yes	no	yes
$q_6$	no	no	yes	yes	yes	no	no

The time for training and inferencing GNN is very small on such a graph (it is small even for large networks as shown in Table 6, but requires to label all the unlabelled data). Querying labelled graph with DLV/Clingo without the ontology takes 0.13-0.15s, which is consistent with Table 7.

Now, let us discuss the main results of the comparison presented in Table 8. We will categorise our observations into distinct aspects for a more comprehensive analysis.

First, if we take *Classroom* ground truth data, almost all the answers will be 'yes', which is a usual case for an OMQ as we saw in Sections IV-A and Section IV-B. The only exception is OMQ  $q_6$  providing a more complex pattern not observed in a given two-community graph.

However, if we try to query *Classroom* data without the ontology (as shown in column Data), we did not find any answer, while both reasoners DLV and Clingo found three 'yes' answers with the help of the ontology. Thus, using reasoners with the help of an ontology provides more consistency with data results compared to not using ontology at all.

Second, all the trained GNN models query labelled data without the use of the ontology, and all provide the answer 'yes', which shows poor quality compared to the results from the reasoners (similar to the results in Section IV-B). However, if we look at one data model presented in the Ground Truth, they have a mistake only in the case of  $q_6$ . Having a larger dataset, probably even "complex" coNP queries, including  $q_6$ , will be answered 'yes' if used with GNN labelling of unlabelled data.

Finally, we observe quite contradicting results leading us to the following two conclusions.

One conclusion is that using machine learning for labelling data produces a number of various structure patterns containing many simple queries directly in the data; thus, machine learning will mostly provide the answer 'yes' for such queries.

Another result lies in the fact that in the real world, every person has mostly fixed node class, and one should be sure whether to consider all the possible data models answering OMQ with respect to the ontology with a covering axiom or directly trying to label the data as close to the real data as possible and then query labelled data with a certain confidence.

Thus, the choice of method is based on the use-case scenario. For e.g., in software verification, you need to check all the possible settings to integrate such a system into production, while in the application of querying social networks, it may be better to get approximate labelling and obtain consistent answers with the one real-world data model. Code of experiments could be found on GitHub.<sup>6</sup>

## V. CONCLUSION

We have considered using a covering axiom as the ontology in the framework of ontology-mediated query answering tasks. During this study, we have received practical experience in applying ontology-based data access with a covering axiom to actual data showing the pros and cons of such an approach.

Initially, we formulated three research questions to evaluate our investigations on how efficient disjunctive datalog reasoners work for the given task and whether it is possible to replace precise computations based on logical deduction with machine learning-based data labelling.

An ontology becomes of current interest when there is missing information in the data or, in our case, when there are unlabelled nodes that could be filled again and produces possible data models with the help of the ontology. That is why, we added uncertainty in data to mask a part of the original data and studied the dependence of datalog and GNN query answering performance versus the size of unlabelled data.

For logic-based reasoners, we have seen that the running time of datalog systems for different sizes of unlabelled data directly depends on the conjunctive query structure. For example, for the reachability-based query  $q_2$ , we can see that the larger the masking size, the faster the answer is searched, as for  $q_5$  belonging to P complexity class, the situation is the opposite. Also, it is clear that the larger query, the greater the running time. In addition, it is important to mention that if the size of unlabelled data is too much, then the ontological approach could fail due to a lack of labels. However, for small graphs (e.g., ego networks), we directly see the advantage of the ontology approach and its benefits in finding missed answers.

Regarding scalability with respect to data size, no general trend shows a dependence between running time and dataset size for all the queries. We can see that for  $q_0$ ,  $q_2$ , and  $q_5$ , in the case of the smaller dataset *Polblogs*, running time is greater, while for the rest queries, the situation is the opposite. However, our datasets have various natures and structural

specifications; that is why, it would be better to provide a proper data scalability analysis for subsets of different sizes from one dataset. Also, as a future work, it is interesting to attempt technically speeding up the search for answers using efficient parallel data processing and analyse scalability performance for the different number of computational clusters [40].

For GNN-based reasoning, we have received that the prediction level for node classification, especially for graphs with negative assortativity, is not enough to replace logic reasoners. However, for large networks, even GNN models with high accuracy will fail, because their node labelling is too excessive with respect to our task, and it is impossible to learn valuable network patterns not impacting almost all simple query patterns used for OMQs.

To sum up, our results highlight the importance of combined analysis of network and query structures and the amount and placement of unlabelled data for choosing between a precise or approximate decision-making method. We have pointed out the limitations of the node classification approach for our problem and the benefits of reasoners and OMQs rewriting in promoting data consistency.

## APPENDIX A

### DATALOG REWRITINGS IN THE SYNTAX OF DLV SYSTEM

```
# T->.->T->.->F
DR0 = P(V) :- T(X), R(X,Y), R(Y,Z), T(Z), R(Z,W), R(W,V),
      A(V).
      P(V) :- P(X), R(X,Y), R(Y,Z), T(Z), R(Z,W), R(W,V),
      A(V).
      P(V) :- P(X), R(X,Y), R(Y,V), A(V).
      G :- T(X), R(X,Y), R(Y,Z), T(Z), R(Z,W), R(W,V),
      F(V).
      G :- P(X), R(X,Y), R(Y,Z), T(Z), R(Z,W), R(W,V),
      F(V).
      G :- P(X), R(X,Y), R(Y,V), F(V).

# T->F
DR1 = P(X) :- F(X).
      P(X) :- A(X), R(X,Y), P(Y).
      G :- T(X), R(X,Y), P(Y).

# T->T->T->F
DR2 = P(V) :- T(X), R(X,Y), T(Y), R(Y,Z), T(Z), R(Z,V),
      A(V).
      P(V) :- P(X), R(X,Y), T(Y), R(X,Z), T(Z), R(Z,V),
      A(V).
      P(V) :- P(Y), R(X,Z), T(Z), R(Z,V), A(V).
      P(V) :- P(Z), R(Z,V), A(V).
      G :- T(X), R(X,Y), T(Y), R(X,Z), T(Z), R(Z,V), F(V).
      G :- P(X), R(X,Y), T(Y), R(X,Z), T(Z), R(Z,V), F(V).
      G :- P(Y), R(X,Z), T(Z), R(Z,V), F(V).
      G :- P(Z), R(Z,V), F(V).

# T<-F->.->T
DR3 = P(X) :- R(X,Y), T(Y), R(X,Z), R(X,V), T(V).
      P(X) :- R(X,Y), T(Y), R(X,Z), R(X,V), P(V), A(V).
      P(X) :- R(X,Y), P(Y), A(Y).
      G :- P(X), F(X).

# T->F->T
DR4 = P(X) :- T(X).
      P(Y) :- P(X), R(X,Y), A(Y), R(Y,Z), P(Z).
      G :- P(X), R(X,Y), F(Y), R(Y,Z), P(Z).

# F->T->.->T
DR5 = P(X) :- T(X).
      P(X) :- A(X), R(X,Y), P(Y), R(Y,Z), R(Z,V), P(V).
      G :- F(X), R(X,Y), P(Y), R(Y,Z), R(Z,V), P(V).
```

<sup>6</sup><https://github.com/Olga3993/MLvsOBDA>

## APPENDIX B

### ORIGINAL DATALOG PROGRAMS IN THE SYNTAX OF CLINGO

```
# t->.->t->.->f
Q0 = g :- t(X), r(X,Y), r(Y,Z), t(Z), r(Z,W), r(W,V),
      f(V).
      t(x);f(x) :- a(x).
      #show g/0.

# t->f
Q1 = g :- t(X), r(X,Y), f(Y).
      t(x);f(x) :- a(x).
      #show g/0.'

# t->t->t->f
Q2 = g :- t(X), r(X,Y), t(Y), r(Y,Z), t(Z), r(Z,W), f(W).
      t(x);f(x) :- a(x).
      #show g/0.

# t<-f->.->t
Q3 = g :- t(X), r(Y,X), f(Y), r(Y,Z), r(Z,W), t(W).
      t(x);f(x) :- a(x).
      #show g/0.

# t->f->t
Q4 = g :- t(X), r(X,Y), f(Y), r(Y,Z), t(Z).
      t(x);f(x) :- a(x).
      #show g/0.

# f->t->.->t
Q5 = g :- f(X), r(X,Y), t(Y), r(Y,Z), r(Z,W), t(W).
      t(x);f(x) :- a(x).
      #show g/0.

# t->f->t->f
Q6 = g :- t(X), r(X,Y), f(Y), r(Y,Z), t(Z), r(Z,W), f(W).
      t(x);f(x) :- a(x).
      #show g/0.
```

## REFERENCES

- O. Gerasimova, S. Kikot, V. Podolskii, and M. Zakharyashev, "On the data complexity of ontology-mediated queries with a covering axiom," in *Proc. 30th Int. Workshop Description Logics*, vol. 1879, A. Artale, B. Glimm, and R. Kontchakov, Eds. Montpellier, France: WordPress, Jul. 2017, pp. 1–12.
- O. Gerasimova, S. Kikot, V. Podolskii, and M. Zakharyashev, "More on the data complexity of answering ontology-mediated queries with a covering axiom," in *Proc. Int. Conf. Knowl. Eng. Semantic Web (Communications in Computer and Information Science)*, vol. 786, P. Rózewski and C. Lange, Eds. Szczecin, Poland: Springer, Nov. 2017, pp. 143–158.
- O. Gerasimova, S. Kikot, A. Kurucz, V. Podolskii, and M. Zakharyashev, "A tetrachotomy of ontology-mediated queries with a covering axiom," *Artif. Intell.*, vol. 309, Aug. 2022, Art. no. 103738.
- H. Ren, W. Hu, and J. Leskovec, "Query2box: Reasoning over knowledge graphs in vector space using box embeddings," 2020, *arXiv:2002.05969*.
- F. Luus, P. Sen, P. Kapanipathi, R. Riegel, N. Makondo, T. Lebesse, and A. Gray, "Logic embeddings for complex query answering," 2021, *arXiv:2103.00418*.
- Z. Zhang, J. Wang, J. Chen, S. Ji, and F. Wu, "ConE: Cone embeddings for multi-hop reasoning over knowledge graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 19172–19183.
- A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati, "Linking data to ontologies," *J. Data Semantics*, vol. 10, pp. 133–173, Jan. 2008.
- D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati, "Tractable reasoning and efficient query answering in description logics: The DL-lite family," *J. Automated Reasoning*, vol. 39, no. 3, pp. 385–429, Oct. 2007.
- F. Baader, I. Horrocks, C. Lutz, and U. Sattler, *Introduction to Description Logic*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
- D. Toman and G. Weddell, "First order rewritability in ontology-mediated querying in horn description logics," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 5, 2022, pp. 5897–5905.
- C. Lutz and L. Sabellek, "A complete classification of the complexity and rewritability of ontology-mediated queries based on the description logic EL," *Artif. Intell.*, vol. 308, Jul. 2022, Art. no. 103709.
- V. Gutiérrez-Basulto, Y. Ibáñez-García, J. C. Jung, and F. Murlak, "Answering regular path queries mediated by unrestricted SQ ontologies," *Artif. Intell.*, vol. 314, Jan. 2023, Art. no. 103808.
- O. Gerasimova, S. Kikot, A. Kurucz, V. Podolskii, and M. Zakharyashev, "A data complexity and rewritability tetrachotomy of ontology-mediated queries with a covering axiom," in *Proc. 17th Int. Conf. Princ. Knowl. Represent. Reasoning*, Jul. 2020, pp. 403–413, doi: 10.24963/kr.2020/41.
- Y. Nenov, R. Piro, B. Motik, I. Horrocks, Z. Wu, and J. Banerjee, "RDFox: A highly-scalable RDF store," in *Proc. 14th Int. Semantic Web Conf.* Bethlehem, PA, USA: Springer, Oct. 2015, pp. 3–20.
- M. A. Musen, "The Protégé project: A look back and a look forward," *AI Matters*, vol. 1, no. 4, pp. 4–12, Jun. 2015.
- M. J. O'Connor and A. Das, "The SWRLTab: An extensible environment for working with SWRL rules in protégé-OWL," in *Proc. 2nd Int. Conf. Rules Rule Markup Lang. Semantic Web*, Jan. 2006, pp. 1–2.
- D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, and G. Xiao, "Ontop: Answering SPARQL queries over relational databases," *Semantic Web*, vol. 8, no. 3, pp. 471–487, Dec. 2016.
- B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, "HermiT: An OWL 2 reasoner," *J. Automated Reasoning*, vol. 53, no. 3, pp. 245–269, Oct. 2014.
- E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL reasoner," *J. Web Semantics*, vol. 5, no. 2, pp. 51–53, Jun. 2007.
- N. Leone, G. Pfeifer, W. Faber, F. Calimeri, T. Dell'Armi, T. Eiter, G. Gottlob, G. Ielpa, C. Koch, S. Perri, and A. Polleres, "The DLV system," in *Proc. 8th Eur. Conf. Logics Artif. Intell.* Cosenza, Italy: Springer, Sep. 2002, pp. 537–540.
- M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub, "Multi-shot ASP solving with clingo," *Theory Pract. Log. Program.*, vol. 19, no. 1, pp. 27–82, Jan. 2019.
- I. Makarov, D. Kiselev, N. Nikitinsky, and L. Subelj, "Survey on graph embeddings and their applications to machine learning problems on graphs," *PeerJ Comput. Sci.*, vol. 7, p. e357, Feb. 2021.
- T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017, *arXiv:1609.02907*.
- L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2017, pp. 1025–1035.
- P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
- H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang, "Geom-GCN: Geometric graph convolutional networks," 2020, *arXiv:2002.05287*.
- D. Bo, X. Wang, C. Shi, and H. Shen, "Beyond low-frequency information in graph convolutional networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, 2021, pp. 3950–3957.
- J. Chen, S. Chen, J. Gao, Z. Huang, J. Zhang, and J. Pu, "Exploiting neighbor effect: Conv-Agnostic GNNs framework for graphs with heterophily," 2022, *arXiv:2203.11200*.
- S. Luan, C. Hua, M. Xu, Q. Lu, J. Zhu, X.-W. Chang, J. Fu, J. Leskovec, and D. Precup, "When do graph neural networks help with node classification: Investigating the homophily principle on node distinguishability," 2023, *arXiv:2304.14274*.
- R. Dolata et al, *Czy Szkoła ma Znaczenie?: Analiza Zróżnicowania Efektywności Nauczania na Pierwszym Etapie Edukacyjnym*, vol. 1, R. Dolata, Ed. Instytut Badań Edukacyjnych, Jan. 2014.
- L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. election: Divided they blog," in *Proc. 3rd Int. workshop Link discovery*, Aug. 2005, pp. 36–43.
- B. Rozemberczki and R. Sarkar, "Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1325–1334.
- I. Makarov, M. Makarov, and D. Kiselev, "Fusion of text and graph information for machine learning problems on networks," *PeerJ Comput. Sci.*, vol. 7, p. e526, May 2021.
- I. Makarov, A. Savchenko, A. Korovko, L. Sherstyuk, N. Severin, D. Kiselev, A. Mikheev, and D. Babaev, "Temporal network embedding framework with causal anonymous walks representations," *PeerJ Comput. Sci.*, vol. 8, p. e858, Jan. 2022.
- I. Makarov, K. Korovina, and D. Kiselev, "JONNEE: Joint network nodes and edges embedding," *IEEE Access*, vol. 9, pp. 144646–144659, 2021.
- O. Platonov, D. Kuznedelev, M. Diskin, A. Babenko, and L. Prokhorenkova, "A critical look at the evaluation of GNNs under heterophily: Are we really making progress?" 2023, *arXiv:2302.11640*.

- [37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [38] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra, "Beyond homophily in graph neural networks: Current limitations and effective designs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 7793–7804.
- [39] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," 2016, *arXiv:1607.00653*.
- [40] H. Mohamed, S. Fathalla, J. Lehmann, and H. Jabeen, "Efficient computation of comprehensive statistical information of large OWL datasets: A scalable approach," *Enterprise Inf. Syst.*, vol. 17, no. 7, Jul. 2023, Art. no. 2062683.



**OLGA GERASIMOVA** received the Ph.D. degree in computer science from the School of Computer Science, HSE University, Moscow, Russia.

Since 2019, she has been a Senior Lecturer with the School of Data Analysis and Artificial Intelligence, HSE University. Since 2020, she has been a Junior Research Fellow with the International Laboratory for Intelligent Systems and Structural Analysis, Faculty of Computer Science, HSE University. Author contribution: datalog experiment

design, coding, paper preparation, and research supervision.



**NIKITA SEVERIN** received the master's degree from the Moscow Institute of Physics and Technology (MIPT), Moscow, Russia. He is currently pursuing the Ph.D. degree in computer science with HSE University, Moscow.

In 2021, he was an Intern with JetBrains Research. Then, he has been a Data Scientist with TradingView. In 2022, he was an Assistant Lecturer of a network science course with MIPT. Author contribution: GNN experiment design, coding, and paper preparation.



**ILYA MAKAROV** received the Specialist degree in mathematics from Lomonosov Moscow State University, Moscow, Russia, and the Ph.D. degree in computer science from the University of Ljubljana, Ljubljana, Slovenia.

Since 2011, he has been a Lecturer with the School of Data Analysis and Artificial Intelligence, HSE University, where he was the School Deputy Head, from 2012 to 2016, and is currently an Associate Professor and a Senior Research

Fellow. He was the Program Director of the BigData Academy MADE, VK, and a Researcher with Samsung-PDMI Joint AI Center, St. Petersburg Department of V.A. Steklov Mathematical Institute, Russian Academy of Sciences, Saint Petersburg, Russia. He is also an Associate Professor with the Moscow Institute of Physics and Technology. He is also a Senior Research Fellow with the Artificial Intelligence Research Institute (AIRI), Moscow, where he leads the research in industrial AI. He became the Head of the AI Research Center and the Data Science Tech Master Program in NLP, National University of Science and Technology MISIS. Author contribution: experiment design, paper preparation, and research supervision.

...