Degree Project in Embedded Systems

Second cycle, 30 credits

Using Graph Neural Network to Analyze Multi-Relational Objects in Dynamic Driving Scenarios

CONGLEI XIANG

Using Graph Neural Network to Analyze Multi-Relational Objects in Dynamic Driving Scenarios

Conglei Xiang

Master of Science Thesis TRITA-ITM-EX 2024:595
KTH Industrial Engineering and Management
Engineering Design
SE-100 44 STOCKHOLM

KTH Industriell teknik och management	**Examensarbete TRITA-ITM-EX 2024:595** **Användning av Grafiska Neurala Nätverk för att analysera multirelationella objekt i dynamiska körscenarier** Conglei Xiang

Godkänt	Examinator	Handledare
2024-10-11	Dejiu Chen	Peng Su
	Uppdragsgivare	Kontaktperson

Sammanfattning

Förståelsen av dynamiska körscenarier innebär betydande utmaningar vid utvecklingen av Automatiserade Körsystem. Automatiserade Körsystem behöver uppfatta viktig information från dynamiska körscenarier för att upprätthålla funktionell säkerhet. Informationen från scenarierna uppfattas genom sensorer (t.ex. kameror, IMU, LiDAR) och är vanligtvis ostrukturerad data. Nuvarande tillvägagångssätt förlitar sig ofta på CNN-modeller för att bearbeta denna data. Men även om CNN-modeller är effektiva på att hantera ostrukturerade sensorinmatningar, har de svårt att fånga de komplexa relationella dynamikerna mellan trafikdeltagare, vilka är avgörande för en djupare förståelse av körmiljön. För att hantera detta problem erbjuder GNN-modeller, som är designade för relationell data, ett lovande alternativ.

Denna avhandling föreslår ett ramverk genom att använda RGCN-modeller, en speciellt designad typ av GNN-modeller för relationell data, för att förstå den relationella datan i dynamiska körscenarier. Specifikt, genom att kategorisera typerna av trafikdeltagare, har ett arbetsflöde utformats för att omvandla ostrukturerad data till spatial relationell data. Genom att analysera ramarna inom en tidsserie med en regelbaserad metod, omvandlas den spatiala relationella datan till spatial-temporal relationell data. Därefter utför RGCN-modellen inferenser om trafikdeltagarnas beteenden och relationerna mellan dem. Resultaten visar att med antagandet av RGCN-modeller blir ramverket enastående när det gäller att klassificera noder och förutsäga deras relationer i en öppen källkodsdataset från verkliga världen. Jämfört med klassiska GCN-modeller uppnår de föreslagna metoderna en förbättring på cirka 10% i prediktionsnoggrannhet, från 0,77 till 0,85 i nodklassificering och från 0,74 till 0,82 i länkprediktion. Dessutom, förutom uppgifterna om nodklassificering och länkprediktion, stödjar denna avhandling också resonering kring okända förhållanden genom att koda kända relationer med en top-1 noggrannhet på 0,78 och hits@2 på 0,91, vilket kan hjälpa till att förbättra förståelsen av riskscenarier.

Nyckelord

Automatiserade Körsystem, Dynamiska Körscenarier, Relationella Grafiska Neurala Nätverk, Multirelationella Data

KTH Industrial Engineering and Management	**Master of Science Thesis TRITA-ITM-EX 2024:595** **Using Graph Neural Network to Analyze Multi-Relational Objects in Dynamic Driving Scenarios** Conglei Xiang	
Approved 2024-10-11	Examiner Dejiu Chen	Supervisor Peng Su
	Commissioner	Contact person

Abstract

Understanding dynamic driving scenarios poses significant challenges in the development of Automated Driving Systems (ADS). ADS need to perceive crucial information from dynamic driving scenarios to maintain functional safety. The information from the scenarios is perceived through sensors (e.g., cameras, IMU, LiDAR) and is usually unstructured data. Current approaches often rely on Convolutional Neural Networks (CNNs) to process this data. However, while CNNs are effective at handling unstructured sensor inputs, they struggle to capture the complex relational dynamics between traffic participants, which are crucial for a deeper understanding of the driving environment. To address this gap, Graph Neural Networks (GNNs), designed for relational data, offer a promising alternative.

This thesis proposes a framework by adopting Relational Graph Convolutional Networks (RGCNs), a special-designed GNNs for relational data, to understand the relational data regarding dynamic driving scenarios. Specifically, by categorizing the types of traffic participants, a workflow is designed to convert unstructured data into spatial relational data. By analyzing the frames within a time series using the rule-based approach, the spatial relational data is converted into spatial-temporal relational data. Next, the RGCNs model infers the behaviours of the traffic participants and the relationships among them. As results, with the adoption of the RGCNs, the framework outperforms in classifying nodes and predicting their relationships in the open-source real-world dataset. Compared to classic Graph Convolutional Networks (GCNs), the proposed methods achieve an improvement of about 10% in prediction accuracy, increasing from 0.77 to 0.85 in node classification, and from 0.74 to 0.82 in link prediction. Furthermore, apart from the tasks on node classification and link prediction, this thesis supports the reasoning of unknown conditions by encoding known relationships with a top-1 accuracy of 0.78 and hits@2 of 0.91, which can help improve comprehension of the risk of the scenario.

Keywords

Automated Driving Systems, Dynamic Driving Scenarios, Relational Graph Convolutional Networks, Multi-Relational Data

Acknowledgments

I would like to thank my supervisor Peng Su, for his guidance, continuous support, and encouragement throughout the duration of this thesis project. I am especially thankful for the many discussions we had, which not only provided me with inspiration but also deepened my understanding of the subject matter, enabling me to approach the research from new and exciting perspectives. I would also like to extend my sincere thanks to my examiner, Professor Dejiu Chen, for taking the time to review my work and for his thoughtful suggestions and comments, which have helped me refine this thesis further.

Furthermore, I would like to acknowledge the support of my family, and friends. Studying in a foreign country has been both challenging and rewarding, and their constant support, encouragement, and companionship have been my source of strength during this journey.

Lastly, I am grateful to everyone who, in one way or another, contributed to the completion of this thesis.

Stockholm, October 2024
Conglei Xiang

Contents

List of Figures

List of Tables

List of acronyms and abbreviations

ADS	Automated Driving Systems
BEV	Bird's Eye View
CNN	Convolutional Neural Network
GAT	Graph Attention Network
GCN	Graph Convolutional Network
GNN	Graph Neural Network
KG	Knowledge Graphs
RGAT	Relational Graph Attention Network
RGCN	Relational Graph Convolutional Network
RNN	Recurrent Neural Network
YOLO	You Only Look Once

Chapter 1

Introduction

This chapter describes the background that the thesis investigates, the motivation and context of the research questions, the purposes and goals of this thesis project, and outlines the structure of the thesis.

1.1 Background

The automotive industry constantly evolves in an era of rapid technological growth, with a special emphasis on developing cutting-edge functions such as Automated Driving Systems (ADS). In pursuing highly dependable and secure automated driving solutions, one of the most significant requirements in the field of automated driving is to identify the surrounding environment of the vehicle accurately, as interactions between vehicles are necessary and pervasive. Understanding dynamic driving scenarios provides crucial information for the ADS to maintain functional safety. Perceiving the information from the scenarios always relies on the in-vehicle sensors, such as RGB cameras, LiDar, Inertial Measurement Unit, *etc*. The information is unstructured data and the common approach to handling the data is learning strategies, *e.g.*, Convolutional Neural Networks (CNNs) for processing the unstructured data like pixel frames, and Recurrent Neural Networks (RNNs) for time-related data.

However, these approaches have some limitations in understanding the scenarios, especially, the behavior of the traffic participants and the relationship between them because they ignore some contextual information. On the one hand, although RNNs are widely used for analyzing and predicting data series, they have limitations in simulating spatial relationships, such as vehicle interaction, and processing image-like data, such as the context of a driving scene. On the other hand, although CNNs have shown success in vision-based prediction tasks, they may struggle with modeling temporal dependencies and handling variable-length sequences, and the interpretability of the learned models can be a challenge (Bharilya and Kumar (2023)).

When considering prediction techniques that take interaction-related factors into account, Graph Neural Networks (GNNs) (Scarselli et al. (2004a))are highly suited, owing to their power in analyzing graph-structured data. Automated driving scenarios often involve complex spatial and temporal relationships among various dynamic and static entities, such as vehicles, pedestrians, traffic signs, and the road itself. Therefore, converting the unstructured data into relational data with a graph structure allows a natural way to capture these relationships. In the graph structure, nodes represent traffic participants, and edges represent interaction relationships. As there are diverse node types and relationship types and they formed heterogeneous graphs, the Relational Graph Convolutional Networks (RGCNs) which is a special type of GNNs designed for multi-relational graph is used in this project.

In this project, a GNN-based system will be built to predict relationships between multiple objects and the behavior of the objects to develop an understanding in the context of automated driving.

1.2 Problem

ADS require the ability to perceive and comprehend crucial information from dynamic driving scenarios to ensure functional safety. Although, CNNs have proven effective in tasks like image recognition, object detection, and scene segmentation, which are essential for understanding the immediate environment around the vehicle. However, there are still challenges in understanding the relationships between different objects, and understanding the context of the scenario, where GNNs outperform. In this project, GNN is adapted to understand the dynamic driving scenarios.

To perceive the surrounding environment, most existing approaches rely on the combined information from various sensors, including cameras, inertial measurement units (IMUs), LiDAR, and other advanced technologies. However, sensors like LiDAR are expensive, whereas cameras offer a much more cost-effective alternative. Therefore, this project aims to explore the possibility of relying solely on video or image data from cameras to achieve accurate recognition of the surroundings. If successful, this approach could significantly reduce the costs of real-world applications while maintaining reliable performance. The video and photo are typical unstructured data. Processing this unstructured data to extract meaningful insights and converting it to the graph-based format which the GNNs can accept is a significant challenge. All that needs to be done is to maximize the retention of the information contained in these data. The first research question that needs to be addressed is:

- How to model the graph-based data in the context of a dynamic driving scenario?

As the dynamic driving data are highly multi-relational, the proposed framework in this project utilizes the RGCNs to process the graph-based data generated. Traditional models, such as Graph Convolutional Networks (GCNs) and other benchmark approaches often struggle with capturing the complex, multi-relational nature of real-world driving scenarios. The RGCNs, designed to handle heterogeneous graph structures and multi-relational data, offers a potential solution to these limitations. Previous works (Mylavarapu et al. (2020a,b); Yu et al. (2022)) illustrate that the RGCNs outperform in dealing with multi-relational data and predicting the nodes and edges. In the dynamic driving scenarios, this work is investigating:

- How much improvement can be achieved in predicting vehicles' on-road behavior utilizing the RGCNs over the benchmark models in the dynamic driving scenarios understanding?

As for multi-relational data, there is a significant need to infer the relationships of unknown or unobserved edges in a graph. There are existing studies in predicting the unknown relationships in Knowledge Graphs (KG), Yang et al. (2015) proposes a bilinear scoring function DistMult, which shows effectiveness in rule reasoning. Motivated by that, this work is using the combination of the RGCN and the DistMult model to implement rule reasoning in the dynamic driving scenarios, thereby enhancing the framework's ability to predict unknown edges and better access the risk. The third question is:

- Given the known relationships within the scenarios, can RGCN+DistMult model be used in reasoning the relationship of unknown edges?

1.3 Purpose

The purpose of the thesis is to develop an advanced machine learning framework combining RGCN to improve the prediction of relationships between traffic participants in dynamic driving scenarios. This research aims to enhance the safety and reliability of ADS by addressing challenges in interpreting complex, dynamic unstructured data from on-road environments.

1.4 Goals

The goal of this project is to understand the dynamic driving scenarios accurately by applying the GNN learning approach. This has been divided into the following three sub-goals:

1. Develop an effective graph-based modeling framework for dynamic driving scenarios:

   This goal addresses the first research question by focusing on how to model graph-based data in the context of dynamic driving scenarios. The objective is to design and implement a method that effectively captures the complex relationships between various traffic participants.

2. Evaluate and improve predictive performance in vehicle behavior analysis:

   The second goal is centered on assessing and enhancing the predictive performance of the RGCNs. This goal aims to quantify the improvements in predicting vehicles' on-road behavior towards node classification and link prediction tasks, providing insights into the advantages of using RGCNs in dynamic driving scenarios.

3. Implement the RGCN + DistMult model for reasoning unknown relationships:

   The third goal focuses on evaluating the effectiveness of the RGCN + DistMult model in reasoning the relationship of unknown edges within the graph. This goal seeks to accurately predict and infer relationships that are not explicitly provided, using the known relationships as a foundation.

1.5 Ethnic and Sustainability

In the development of ADS, it is crucial to address ethical considerations and sustainability to ensure that the technology not only enhances mobility but also aligns with societal values and long-term environmental goals.

Ethic: The most significant concern is ensuring the safety and reliability of automated driving systems. The method used in this project also contributes to technological advancement in industry and infrastructure areas. The system is repeatedly tested and validated to reduce errors and improve its robustness. Besides, all the data collected are in the proper ways, and all the resources used in this project are in a correct way (*e.g.,* by citation).

Sustainability: This work aims to improve the performance of the ADS by enhancing the ability of it to perceive surroundings. By developing a more in-depth study, different complex scenarios may be supported in the future, which directly affect urban and community life.

1.6 Structure of the thesis

Chapter 1 provides an overview of the general background, research problem, motivation, and objectives. It also outlines the contributions and the ethnic and sustainability of this work.

Chapter 2 presents the relevant theoretical background of the thesis, including Driving Scenario Understanding, key concepts of Graph Neural Networks, Graph-Based Rule Reasoning, and some previous work that is similar to this work.

Chapter 3 presents the research methodology. It covers the research process, data collection from both image and video datasets, data preprocessing techniques, the statistical methods and evaluation metrics employed in this work.

Chapter 4 explains the detailed framework of this research, including the design and implementation of the RGCN model for node classification and link prediction. The chapter also discusses the use of rule reasoning in improving the model's value of practical application.

Chapter 5 presented the experimental setup, including the datasets, evaluation metrics, and baseline models. Additionally, the results of the experiments are analyzed and compared to demonstrate the effectiveness of the proposed methods. The performance of the models and the implications of the findings are also analysed.

Chapter 6 summarizes the key findings of the research and reviews the contributions of the thesis. Limitations of the current approach and potential areas for future work are also discussed.

Chapter 2

Background

This chapter introduces the foundational concepts and related work essential for understanding this study. It begins by exploring driving scenario analysis through object tracking. Then it covers multi-relational graphs and their application in modeling complex interactions among traffic participants. It proceeds to discuss GNNs, including GCNs and RGCN, highlighting their difference in handling multi-relational data. Finally, it reviews previous work in ADS and GNN-based relational data analysis, setting the stage for the research presented in this thesis.

2.1 Driving Scenario Understanding

2.1.1 Object Detection

Since the predictions of this project are all based on the graph structure, and the raw dataset accessed is all in the form of RGB-based pixel-level videos or photos captured by the monocular camera. In order to subsequently apply the datasets in the RGCNs, it is crucial to extract the traffic participants in the datasets to get the relationship between them and get the useful features. In this work, the module implemented for object detection and tracking is You Only Look Once (YOLO)v8 where it could identify vehicles in the scene.

YOLOv8 is the newest model in the YOLO (Redmon et al. (2016)) algorithm series – the most well-known family of object detection and classification models in the Computer Vision field. With the latest version, the YOLO legacy lives on by providing state-of-the-art results for image or video analytics, with an easy-to-implement framework. In addition, since the project needs to capture the dynamic relations of the objects – to identify and compare the same entity between different frames, the object tracking function is critical. Object tracking not only identifies the location and class of objects within the frame but also maintains a unique ID for each detected object as the video progresses. Thus, among the many modes contained in YOLOv8, the Tracking Framework is chosen to meet the requirements.

2.1.2 Multi-Relational Data in Dynamic Driving Scenario

Graphs are becoming a popular choice of data structure when dealing with unstructured data and modeling irregular domains. Graphs are composed of nodes and edges. Nodes are denoted as entities, and edges are denoted as relationships between the entities. However, most of the graphs contain only one node type and a single type of relationship, and they are homogeneous graphs (shown as Figure 2.1).

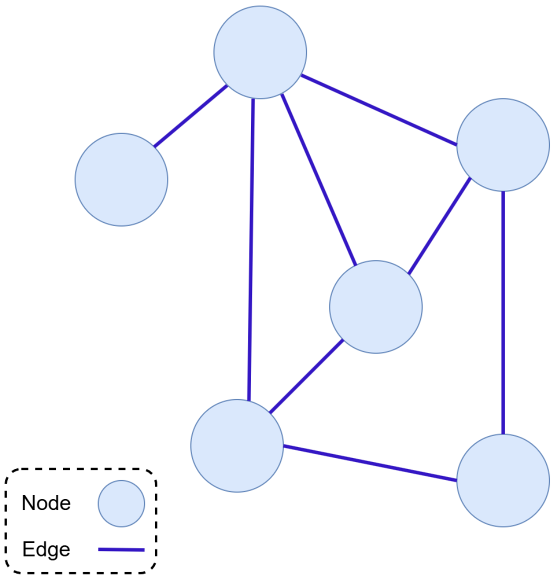


Figure 2.1: Homogeneous Graph Structure

In contrast, heterogeneous graphs can illustrate multiple relationships, which means they can have various node and edge types, as shown in Figure 2.2. Edges between nodes of different types represent different kinds of relationships (*e.g.,* the edge between *Node 1* and *Node 2* is Type 2, and the edge between *Node 1* and *Node 5* is Type 4), and even edges between the same type of nodes can represent different kinds of relationships (*e.g.,* edge between

Node 1 and *Node 2* is Type 2, and the edge between *Node 2* and *Node 3* is Type 3).

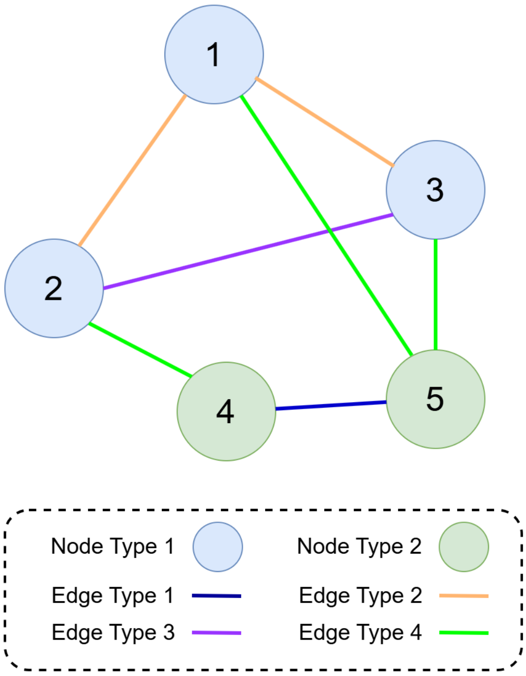


Figure 2.2: Heterogeneous Graph Structure

The most typical and widely-used multi-relational graphs are in KG because the knowledge has various and complex connections between the entities (Li et al. (2022)). Inspired by the KG, the dynamic driving scenario is suitable for being modeled as a multi-relational graph. Because there are several types of traffic participants, *e.g.*, vehicles, and pedestrians, which are denoted as nodes, and there are also several types of spatial-temporal relationships between the traffic participants, which are denoted as edges (Mylavarapu et al. (2020a,b)). The relationships are human-understandable, *e.g.*, *Moving Away*, *Moving Near*, which can help analyze the behavior and comprehend the scenarios.

2.2 Graph Neural Network

In recent years, while the requirement of dealing with the data in the non-Euclidean domains is rapidly booming, the concept of the GNN has been proposed to address the problem (Scarselli et al. (2004b)). Traditional neural networks, such as RNNs and CNNs, are usually adopted to process the data in the Euclidean space (*e.g.*, images, text, and videos) (Wu et al. (2021)). However, in many tasks, dealing with the data in a regular grid, like implementing the CNNs in traffic scenario frames which at the pixel level, may lose some relative information between the objects. GNN is a kind of learning-based method that operates in the non-Euclidean domains, which has been receiving increasing attention recently and applied to different fields. As GNN is based on this unique non-Euclidean data structure, it can capture more relative information and efficiently share the information across the graph. Depending on the various prediction tasks, it can be categorized as node classification, link prediction, and clustering.

2.2.1 Graph Convolutional Network

As CNNs have successfully processed the data with a regular grid, it is intuitive to define the convolution operation in the non-Euclidean domains according to Liu and Zhou (2022). The basic CNNs can learn from fixed-size kernels which can scan every pixel in the images and combine the surrounding information. Convolution operations enable the network to extract more high-level features from the data and attain significant expressing ability. Aiming to generalize the convolution operations on graphs, GCNs were first proposed in Bruna et al. (2014). The convolution operation is defined in the Fourier domain by computing the eigen decomposition of the graph Laplacian.

Most of the existing work is based on the GCN framework proposed in Kipf and Welling (2017). There are some promotions and simplifications in this framework which in many cases allow both for significantly faster training times and higher predictive accuracy, reaching state-of-the-art classification results on a number of benchmark graph datasets.

2.2.2 Relational Graph Convolutional Network

Although the existing GCNs have achieved good performance in many cases, the shortage is obvious. It focuses only on homogeneous graphs and learning node representations without considering the various relationships on edges. Thus, GCNs would not be able to directly implement the prediction of edge relationships. The RGCNs, proposed in Schlichtkrull et al. (2017), can be seen as an extension of the traditional GCN, which is applied to modeling relational data in heterogeneous graphs, specifically to link prediction and node classification tasks. Unlike the traditional GCNs, RGCNs consider the different types of relationships in the graph. By incorporating multiple types of relationships, RGCNs are well-suited for tasks like node classification and link prediction, which are essential for understanding the interactions and behaviors of traffic participants in dynamic driving environments.

The following sections illustrate how RGCNs can be applied to tackle these tasks:

- **Node Classification**

  A node classification task predicts an attribute of each node in a graph. For instance, labeling each node with a categorical class (binary classification or multiclass classification), or predicting a continuous number (regression). It is supervised or semi-supervised, where the model is trained using a subset of nodes that have ground-truth labels.

  The process is shown in Figure 2.3. First, the data is input into the model, including the graph structure, node features, node labels, and edge attributions. Then, the model will implement forward propagation through multiple RGCN layers to update node features, and output node class probabilities through the final layer. In the last step, cross-entropy loss on all labeled nodes will be minimized, and update model parameters with an optimizer.

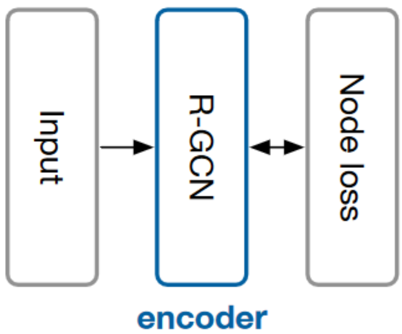


Figure 2.3: RGCN model for Node Classification Task (Schlichtkrull et al. (2017))

- **Link Prediction**

  Link prediction deals with the prediction of edges which can be denoted as a structure of triple *(subject, relation, object)*. When the edges of the graphs are incomplete, the task is to predict the most possible edges that are unknown relying on the known graph structure, node features, labels, and edge attributions.

In order to tackle the link prediction problems, an encoder-decoder structure is proposed as shown in Figure 2.4, which is composed of an entity encoder and a scoring function decoder. The encoder maps each entity to a real-valued vector. The decoder assigns scores to the *(subject, relation, object)*-triplets relying on the vertex representations. In that case, the link prediction task is converted to a ranking task. Scores are computed for the correct entity and all the corrupted entities in the dictionary and are ranked in descending order. With the scores ranking, the edges can be determined how likely those edges are to belong to the relationships.

Many studies were carried out on the scoring function for relational triplets. *E.g.*, Distance proposed by Bordes et al. (2011), Single Layer proposed by Glorot et al. (2013), TransE proposed by Bordes et al. (2013), NTN proposed by Socher et al. (2013), *etc*. However, the DistMult, proposed by Yang et al. (2015), is widely used because of its simple mathematical structure and competitive results. It performs well on both standard link prediction benchmarks when used on its own (Yang et al. (2015)), and RGCN model (Schlichtkrull et al. (2017)).

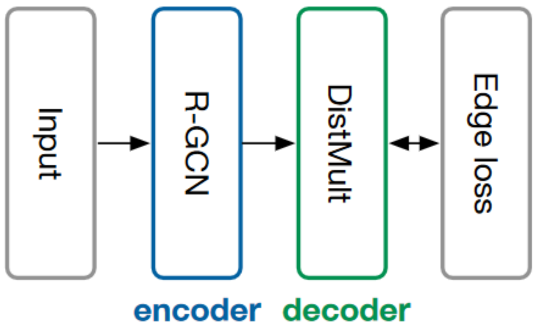


Figure 2.4: RGCN model for Link Prediction Task (Schlichtkrull et al. (2017))

2.3 GNN-Based Rule Reasoning

Logical rule reasoning is a concept that has always existed in knowledge graph reasoning. For example, in Figure 5.5, the triplets denoted with the black line – *("Taylor Swift", "born_in", "Pennsylvania")*, and *("Pennsylvania",* *"located_in", "the USA")* are the information already existed in the graph. Given these facts, the triplet *("Taylor Swift", "country_of_birth", "the USA")* can be inferred according to the logic of the given relationships. Such logical rules can help complete the existing graph.

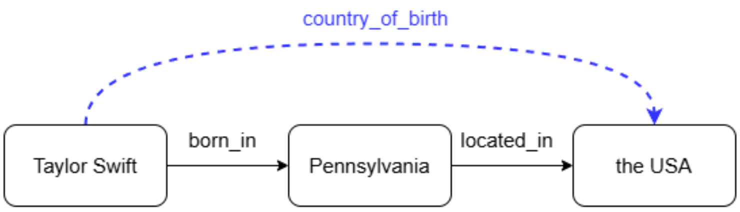


Figure 2.5: Example of the rule reasoning in graph

As spatial-temporal relationships have a strong logic, rule reasoning can be implemented in this work to infer the unknown relationships between traffic participants in dynamic scenarios. For example, in Figure 2.6, given that

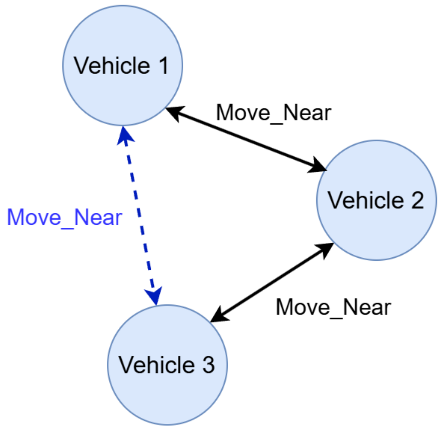


Figure 2.6: Example of the rule reasoning in dynamic scenario graph

Vehicle 1 is moving near Vehicle 2, and Vehicle 2 is moving near Vehicle 3, the unknown relationship between Vehicle 1 and Vehicle 3 has to be "moving_near" instead of "moving_away". This approach can help improve the robustness of link prediction and provide a better understanding of the scenarios.

Yang et al. (2015) proposed the detailed algorithm of the rule reasoning based on the DistMult Scoring Function. The relationship illustrated in Figure

2.5 and 2.6 can be modeled as Horn rules that consist of a head relation H and a sequence of body relations $B_1, ..., B_n$. For simplicity, only Horn rules of length 2 is considered:

$$B_1(a, b) \wedge B_2(b, c) \Rightarrow H(a, c) \tag{2.1}$$

where a, b, c are variables that can be substituted by entities, and the body of the rule can be viewed as the composition of relations B1 and B2.

The relation composition is modeled as the multiplication of the two relation embeddings when using the DistMult Scoring Function. The composition results in a new embedding that lies in the same relation space.

2.4 Previous Work

This section briefly reviews previous work in ADS and the work implementing GNN approaches.

2.4.1 AI-Based Methods in ADS

As the traffic scenarios are complicated and non-determined, it is always a big challenge for ADS to ensure safe and stable operation. To ensure the robustness and safety of the system, AI approaches are widely used. On one hand, they help with risk management, and on the other hand, they assist in the perception of driving scenarios and understanding of the surrounding environment.

Fault Injection is the main approach widely used to decrease the risk of the system. Su et al. (2023b) proposed a simulation-aided approach to enhance the safety analysis of ADS by generating operational data, and injecting faults into high-dimensional sensor data. Their framework also implemented Variational Bayesian methods to estimate operational conditions, demonstrated through an Autonomous Emergency Braking system case study in various weather conditions. Su et al. (2023a) proposes methods for extracting operational conditions, software services for fault injection and data synthesis, and support for operation simulation and data analysis. Su and Chen (2022) introduced a Fault Injection method, injecting layer- and neuron-wise faults into neural networks. The impacts are quantified using a probabilistic criterion based on Kullback-Leibler Divergence, demonstrated through tests with AlexNet.

2.4.2 GNN Approaches in Relational Data

Real-world data are not universally stored as unstructured data but in the interlinked structures of internet pages, social networks, knowledge graphs, biological, chemical, *etc*. These are inherently relational data that are naturally stored in their structured form of graphs. Thus, GNNs are used in processing the relational data.

For example in KG, GNNs meet the requirements of a knowledge graph for learning the attribute features and structural features of entities and relationships, implemented in 4 tasks – node classification, link prediction, knowledge graph alignment, and knowledge graph reasoning (Wu et al. (2021)).

GNNs also have state-of-art applications in Human Activity Recognition (HAR) to support analyzing individuals' Activities of Daily Living (ADL). In Su and Chen (2024), human-object interactions are modeled as relational graphs, and they proposed a framework using GNN to analyze human–object interactions for more effectively recognizing daily activities.

In recent years, studies have been various in GNN-based Driving Scenario Understanding because the driving scenarios have strong relationships within the traffic participants and can be naturally modeled as relational data. Previous work can be categorized into trajectory analysis, operational risk identification, and behavior prediction:

(1) Trajectory analysis aims to predict the future trajectory from a given trajectory history. It relies on temporal-spatial information to cope with decision-making in ADS. Xu et al. (2022) proposed a network for extracting comprehensive spatial-temporal feature representations.

(2) Operation risk identification focuses on assessing the subjective risk of driving maneuvers from the scene graphs. Yu et al. (2022) proposed a framework that combined a Multi-Relation Graph Convolution Network, a Long-Short Term Memory Network, and attention layers to identify high-risk entities from the video clips.

(3) Works in behavior prediction are the most similar to this project. Mylavarapu et al. (2020a,b) modeled the traffic scenarios as a relational graph, with the primary task being the prediction of relationships among traffic participants. In this context,GNNs are used to extract node and edge information to predict dynamic relationships between objects. However, the spatial relationships rely on the landmarks (*e.g.,* lane marks), introducing additional requirements for the dataset. In Su et al. (2024), the proposed framework can predict spatial-temporal relationships with both regression and

classification results to better recognize the operational contexts.

Chapter 3

Research Methodology

The purpose of this chapter is to provide an overview of the research methodology in this thesis. Section 3.1 describes the research process. Section 3.2 focuses on the data collection and datasets used for this research. Section 3.3 briefly describes how the source data is processed into graph-structured data that can be handled by the model. Section 3.4 illustrates the quantitative statistical method used to analyse the experiment data results. Finally, Section 3.5 describes the evaluation metrics selected to evaluate the performance of the proposed framework.

3.1 Research Process

In this section, the research process of this thesis will be illustrated. As shown in Figure 3.1, the complete process contains 7 steps, defining research fields and directions, literature study, research design, data collection and processing, model construction, experiment, and results analysis.

Firstly, motivated by the previous work, the idea about utilizing GNNs in ADS is come up with. In Which field that the thesis will investigate is determined in a more detailed way. Besides, specific research questions and goals are initially defined.

Secondly, during the literature study, published academic papers, books, journal articles, *etc.,* are read and analyzed to keep abreast of current knowledge and research advances in the field of applying GNNs, particularly RGCNs in dynamic driving scenarios.

Thirdly, it was the step to make a detailed research plan. In this period, the selection of appropriate research methodology and the methods of data analysis based on research questions and research objectives are designed.

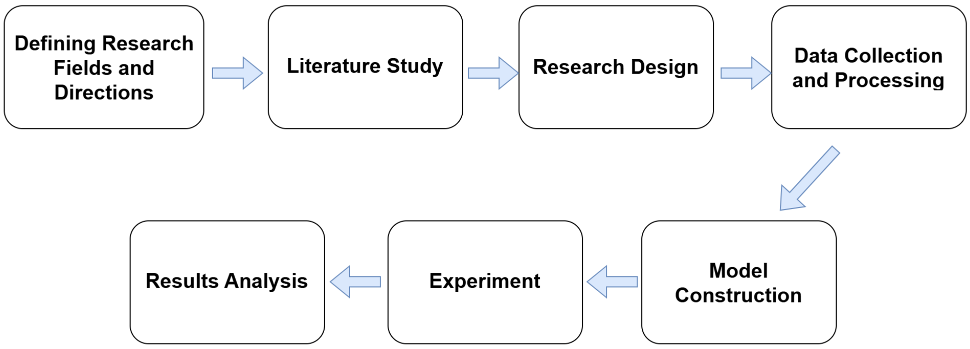


Figure 3.1: Research Process

Specific milestones are planned according to the research process and the time requirements.

After that is the data collection, proper datasets are chosen and processed to match the format of the RGCNs. In this thesis, the datasets are all open-source real-world driving datasets.

Subsequently, the framework of the project is constructed. In this step, node classification and link prediction tasks are first implemented with RGCNs. Attempts and extensions have been made to the model in order to gain more accurate results and to investigate deeper. Apart from that, some adjustments have been made to the initial research questions according to the practical situation.

In the experiment step, the baseline models are constructed as a benchmark, and compared to the model proposed in this work to illustrate the performance and effect of the proposed method. The results are gained by quantitative measures with evaluation metrics as they provide insights into how well the model is performing and help in comparing different models or algorithms.

Last but not least, the results are analyzed and interpreted with statistical methods. The findings are linked to the literature review and existing theories to explore the significance, limitations, and potential implications.

3.2 Data Collection

3.2.1 Image Dataset – *KITTI*

KITTI dataset by Geiger et al. (2012) is one of the most commonly used datasets internationally for automated driving scenarios, developed by the Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago. The data acquisition platform for the KITTI dataset is fitted with two greyscale cameras, two colour cameras, a Velodyne 64-line 3D Lidar, four optical lenses, and a GPS navigation system as shown in 3.2, and the system recorded the urban traffic scenes for up to 6 hours. The dataset is a collection of images and LIDAR data used in computer vision research, such as stereo vision, optical flow, visual odometry, 3D object detection, and 3D tracking. The dataset is publicly available and can be freely downloaded. In this work,

Figure 3.2: KITTI Recording Platform (Geiger et al. (2012))

we used the *Object Detection Evaluation 2012* Dataset. The object detection and object orientation estimation benchmark consists of 7481 training images and 7518 test images, comprising a total of 80.256 labeled objects. All images are colored and saved as PNGs. As our work is only based on monocular vision, only the color images from the left are adapted. All the images are from ego view.

3.2.2 Video Dataset – *PIE*

PIE is a dataset for studying the traffic situation. Apart from on-road vehicles, PIE also contains various pedestrians as traffic participants. PIE contains over 6 hours of footage recorded in typical traffic scenes with an on-board camera.

It also provides accurate vehicle information from OBD sensor (vehicle speed, heading direction and GPS coordinates) synchronized with video footage. Rich spatial and behavioral annotations are available for pedestrians and vehicles that potentially interact with the ego-vehicle as well as for the relevant elements of infrastructure (traffic lights, signs and zebra crossings). There are over 300K labeled video frames with 1842 pedestrian samples making this the largest publicly available dataset for studying pedestrian behavior in traffic.(Rasouli et al. (2019))

3.3 Data Preprocessing

3.3.1 Object Detection

In this part, the YOLOv8 model is employed. It is a state-of-the-art object detection model in the YOLO family. In order to know the spatial-temporal relationships among the objects, the model must be able to detect the same object through all the pixel-level frames which is the trajectory of the objects. The object tracking module of YOLOv8 model can fulfill this need with high confidence. Each frame from the videos was processed by YOLOv8, resulting in bounding box coordinates (e.g. *x_left*, *x_right*, *y_top*, *y_bottom*), object IDs and classification labels for each detected object.

3.3.2 Scene Graph Construction

The bounding box coordinates can indicate the spatial location. To analyze the objects' movements in a real-world context, the pixel coordinates are converted into Bird's Eye View (BEV) coordinates using a transformation matrix. The transformed coordinates can be applied to examine the spatial-temporal relationships between the objects more accurately. The relationship is assessed whether the objects were moving closer together, moving further apart, or keeping the same distance. These provide insights into traffic dynamics.

3.4 Statistical Method

The statistical method used when processing the results data is calculating the mean. It provides a single value that summarizes the entire dataset, making it easier to understand and communicate the typical value in a dataset.

The comparison experiments are replicated several times and the factors are controlled to reduce errors. Averaging helps smooth out these random fluctuations, providing a clearer view of the underlying trends or effects.

3.5 Evaluation framework

Enabling more accurate predictions, faster insights, and improved decision-making, quantitative methods are utilized in results analysis to interpret the performance of the proposed framework in this thesis. Quantitative analysis involves using mathematical and statistical methods to analyze data and make informed decisions based on numerical evidence. In order to assess the performance and effectiveness, the evaluation matrices are adopted as a quantitative measure. According to the experiment and the literature, proper evaluation matrices have been chosen to provide objective criteria to evaluate a model as thoroughly as possible. In this thesis, as the main tasks are divided into node classification and link prediction, the evaluation metrics are varied depending on the task.

(1)Node Classification Classification Accuracy and F1-Score are adopted. In the classification tasks, a confusion matrix is widely used, which is a matrix that summarizes the performance of a machine learning model on a set of test data. It is a binary classification task in this thesis, which consists of “positive” and “negative” classes. Then the Confusion Matrix has four essential components as in Figure 3.3:

(1) True Positives (TP): Number of samples correctly predicted as “positive.” (2) False Positives (FP): Number of samples wrongly predicted as “positive.” (3) True Negatives (TN): Number of samples correctly predicted as “negative.” (4) False Negatives (FN): Number of samples wrongly predicted as “negative.”

Classification accuracy is a fundamental metric for evaluating the performance of a classification model, providing a quick snapshot of how well the model is performing in terms of correct predictions. It is calculated as the ratio of correct predictions to the total number of input Samples.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.1}$$

The accuracy metric computes how many times a model made a correct prediction across the entire dataset, and highly demanding whether the classification of the dataset is balanced or not. However, the dataset from

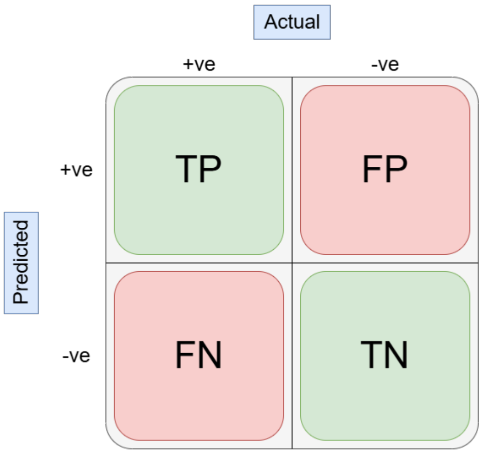


Figure 3.3: Structure of a confusion matrix

the real world is not that balanced. The F1 score combines precision and recall using their harmonic mean, and maximizing the F1 score implies simultaneously maximizing both precision and recall. Thus, the F1 score has become the choice of researchers for evaluating their models in conjunction with accuracy. The mathematical definition is shown as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.3}$$

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

(2)Link Prediction According to Schlichtkrull et al. (2017); Yang et al. (2015), hits@k is a commonly used evaluation metric in link prediction tasks. It assesses how well the model ranks the true missing links compared to the false ones. The metric calculates the percentage of correct predictions within the top k ranked links. The mathematical definition is shown as follows:

$$\text{Hits@k} = \frac{1}{N} \sum_{i=1}^{N} 1[\text{rank}(i) \leq k] \tag{3.5}$$

where N is the total number of predictions (or test cases), $1[\cdot]$ is the indicator function that equals 1 if the rank of the correct entity rank(i) is within the top k, and 0 otherwise. A higher Hits@k value indicates better performance in predicting the relationships.

Chapter 4

Details of Methodological Framework

This chapter delves into the comprehensive explanation of the framework employed in the study, focusing on the methods and processes used to achieve the research objectives. This chapter provides a detailed description of how the multi-relational graph is constructed, how the vehicle direction prediction is implemented, and how the multi-relationships among objects are predicted.

4.1 Framework Overview

To understand the dynamic driving scenarios accurately and obtain a reliable prediction of the current risk level, a framework is proposed based on the multi-relational objects. This framework, as illustrated in Figure 4.1, is a methodical pipeline that could be divided into three stages:

- The first stage is understanding the dynamic driving scenarios and extracting relational data from the pixel-level datasets. The YOLO network is adopted to track the dynamic objects including vehicles and pedestrians within from the time-series datasets. The coordinates of the bounding boxes, the class of each object, and their specific IDs are captured in this stage, and these are taken as spatial information. All the spatial information is compared within a time period to figure out the spatial-temporal relationship.
- In the second stage, multi-relational spatial-temporal graphs are constructed according to the information gathered from the previous stage. The edges of the graphs represent the spatial-temporal

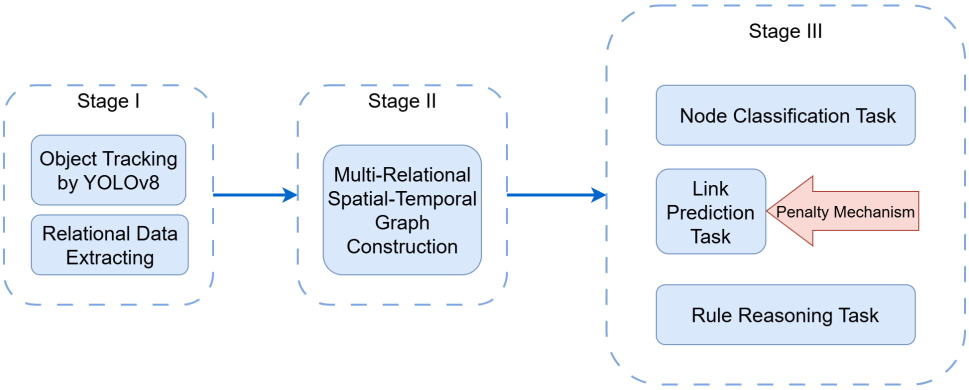


Figure 4.1: Overview pipeline of the project

relationships, and the nodes represent the traffic participants, containing the labels and the features.

- The last stage is adopting RGCNs to the graphs. In this stage, node classification, link prediction, and rule reasoning tasks are implemented. In the node classification tasks, the situation of the nodes which indicates vehicles moving directions are predicted, which illustrates whether the car is in the same lane or in the opposite lane to the ego car. As for the link prediction tasks, the relationships between all the vehicles and the relationships between each pedestrian and the ego vehicle are labeled and predicted. These prediction results illustrate the dynamic behavior of the surrounding on-road objects. Furthermore, penalty mechanism is injected into this task to improve the accuracy of the prediction. In the rule reasoning task, the unknown relationships are predicted to enhance the comprehension of the scenarios to assess the risk level between the on-road objects.

4.2 Multi-Relational Graph Construction

In this section, the construction of a multi-relational graph of the dynamic driving scenario is detailed. The raw datasets are pictures or videos from the on-vehicle monocular camera of the dynamic driving scenarios. The purpose of this section is to convert these unstructured datasets into a graph structure, and the process is shown in Figure 4.2.

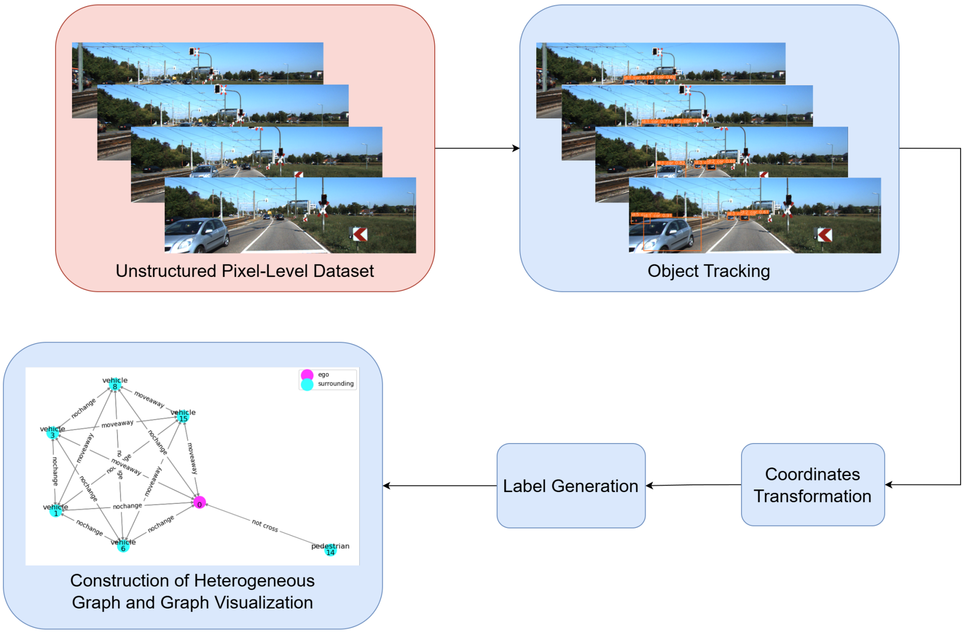


Figure 4.2: Process of the Multi-Relational Graph Construction

4.2.1 Object Tracking

In order to recognize the object in the scenarios, the tool YOLOv8 used for semantic recognition is utilized. There are several pre-trained models in YOLOv8 framework for various tasks. In this project, the tracking framework is selected after some attempts. The basic object detection task can recognize each frame with high confidence and output the coordinates and class of the object bounding box. However, as the project aims to deal with dynamic tasks within a time series, basic object detection has a shortage that it cannot match the same object in different frames. Thus, the object tracking framework, based on the object detection model adding the Ultralytics trackers better suits the goals. The output from Ultralytics trackers is consistent with standard object detection but has the added value of object IDs. This makes it easy to track the same object in dynamic streams dataset and perform subsequent analytics.

The YOLOv8 object tracking framework recognizes objects with bounding boxes and outputs the bounding boxes' IDs, location coordinates (x_left, x_right, y_top, y_bottom), and objects' classes (dynamic and static objects, *e.g.,* car, people, truck, traffic light).

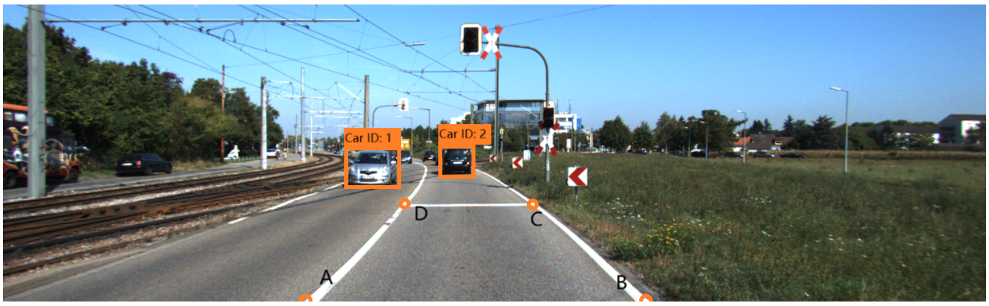
4.2.2 Coordinates Transformation

Next, a transformation for the coordinates to BEV is operated. Because it is more intuitive to get the distance of each dynamic object in the scene with BEV, and more conducive to the determination of the relationship between the objects and to set the rule-based threshold in the subsequent steps. The transformation relies on OpenCV's perspective transformation library to approximate each object's location (Yu et al. (2020)). Although this method is not as precise as using the calibration parameters of the camera, it can provide enough relative information required in the tasks and the process is much easier:

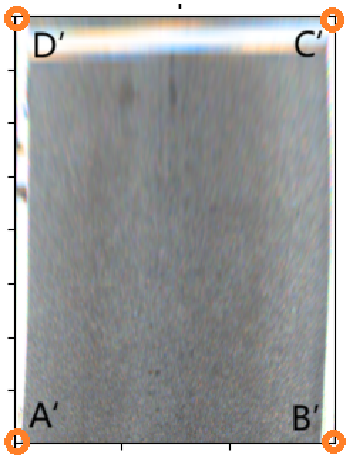
1. Make sure all the frames in the dataset are in the same size.
2. Choose a frame from the dataset, and set 4 points in the frame image, and get their coordinates. *E.g.,* in Figure 4.3a, the four points – A, B, C, D frame an area of the roadway.
3. Set the transformed coordinates of these four points. *E.g.,* The area framed by these four points is approximated as a rectangle from the BEV. The transformed coordinates of A', B', C', D' are set to be [0,400], [300,400], [300,0], [0,0] as shown in Figure 4.3b and 4.3c.
4. OpenCV helps to calculate the transformative matrix. With the matrix, all the coordinates data in the same dataset can be transformed into BEV. The transformed BEV which illustrates the locations (center point coordinates of the bounding box) of the objects in Figure 4.3a is in Figure 4.3c. As the ego vehicle does not appear in the tracking process, the coordinates of the bottom center of each frame are defaulted to ego car location coordinates ([150,400] in this example) and its ID is always 0. The green point represents the location of the ego vehicle, the blue point represents car number 1, corresponding to the silver car in Figure 4.3a, and the orange point represents car number 2, corresponding to the black car in Figure 4.3a.

4.2.3 Generating Node and Edge Labels

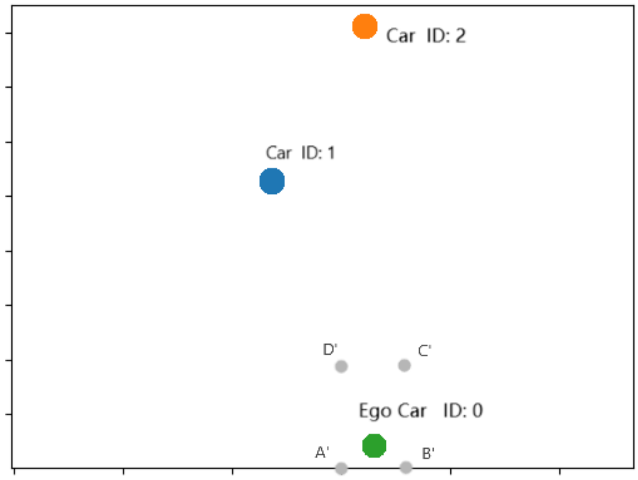
Subsequently, the node labels and edge labels are identified with a rule-based method. In this step, the coordinates of the static objects like traffic lights are ignored according to the class from the object tracking results because the final multi-relational graphs only contain the dynamic objects.



(a) Frame before transformation



(b) Transformed output of the selected area



(c) Location of transformed objects

Figure 4.3: An example process of transformation to BEV

The node labels are classified into 2 classes – *'same' and 'opposite'*, which represent the direction of the surrounding vehicles. The reference object in the 'same' and 'opposite' directions is the ego-vehicle. According to the raw frame datasets, the area of the same direction lanes and the opposite direction lanes are labeled with a threshold of the coordinates in each scenario. Therefore, the surrounding vehicles' bounding box coordinates are quantitatively compared with the ego vehicle's location. If the surrounding vehicle's location is in the area of the same direction lane, then the node label of this object is 'same' and encoded as 1; If the surrounding vehicle's location is in the area of the opposite direction lane, then the node label of this object is 'opposite' and encoded as 0.

The edge labels are classified into 5 classes – *'movenear', 'moveaway', 'nochange', 'cross', 'notcross'*. These five classes are all spatial-temporal relationships, and a sliding window is set to check the objects' location relationships in a fixed time period. They offer a structured approach to analyzing and understanding the spatial-temporal relationships between traffic participants in automated driving scenarios. The first three classes represent the relationship between each two vehicles, the latter two represent the relationship between each pedestrian and the ego vehicle. As for the relationship between each two vehicles, the distance between each two vehicles at each moment is calculated with their location coordinates. In a sliding window, the amount of change in distance of the two vehicles is calculated and compared with a fixed threshold to identify whether the two vehicles are moving near (labeled as 1), moving far away from each other (labeled as 2) or having no obvious change in their distance (labeled as 0). The relationship between each pedestrian and the ego vehicle is determined in a similar way, but relies only on the horizontal location coordinate. As the ego vehicle has fixed coordinates, calculating the amount of change in the pedestrian's horizontal coordinate at each moment in the sliding window and comparing it to a threshold value would give the result of whether the pedestrian is crossing (labeled as 3) or no obvious crossing (labeled as 4) behavior.

4.2.4 Construction of Heterogeneous Graph and Graph Visualization

The last step of construction of the multi-relational graphs is to save the data into **torch_geometric.data.HeteroData** format since the subsequent GNN can directly take these data as input. All the traffic participants are represented as nodes, which contain the bounding box center point's coordinates as features, and the vehicle nodes contain labels of their moving directions. The relationships are represented as the edges, containing the encoded relationship labels. Additionally, to enhance its visualization, the NetworkX tool is utilized to output some graphs as shown in Figure 4.4, so that they can be compared

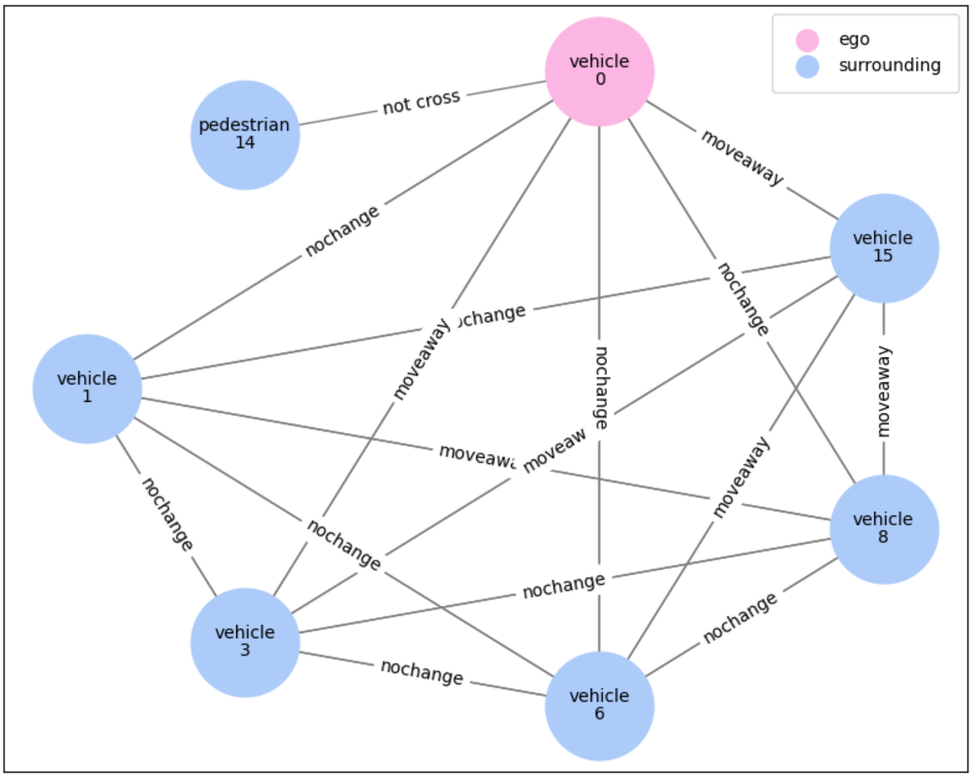


Figure 4.4: Example visualized graph with NetworkX

with the images or videos in the original dataset, ensuring to some extent the reliability of the generated graphs.

4.3 Implementation of Vehicle Direction Prediction

The spatial-temporal representations of the traffic participants are obtained in the last section by modeling the temporal dynamics of their spatial relations with other objects in the scenarios over time. To predict the situation of each node, an RGCN framework is proposed to learn relevant information from different spatial-temporal relations. The multi-relational graph is defined as $G = (V, E)$ with vertex set V and edge set E. $E_{i,j} \in R_d$ is an edge between node i and j. The i^{th} node feature obtained from a graph convolution over relation, r in l^{th} layer is defined as:

$$h_r^l[i] = \sum_{j \in \mathcal{N}_r[i]} \frac{1}{c_r[i]} W_r^l h^{l-1}[j] \tag{1}$$

where, $\mathcal{N}_r[i]$ denotes the set of neighbor nodes for v_i under relation r, $\mathcal{N}_r[i] = \{j \in V \mid E_{j,i} = r\}$, and $c_r[i] = |\mathcal{N}_r[i]|$ is a normalization factor. Here, $W_r^l \in R^{d' \times d}$ is the weights associated with relation r in the l^{th} layer of RGCN; d', d are the dimensions of the $(l-1)^{th}$ and l^{th} layers of RGCN. Neighborhood information aggregated from all the relations are then combined by a simple summation to obtain the node representation as follows:

$$h^l[i] = \text{ReLU}\left(W_s^l h^{l-1}[i] + \sum_{r \in R_d} h_r^l[i]\right) \tag{4.1}$$

where, the first terms correspond to the node information (self-loop) and $W_s \in R^{d' \times d}$ is the weight associated with self-loop. To account for the nature of the entity (active or passive), we learn entity embeddings, $\mathcal{E}_i \in R^{|O| \times d}$ for node v_i, where $O = \{\text{Vehicles}\}$. The input to the first layer of the RGCN, $h^0[i]$, is the embedding $\mathcal{E}_i$ based on the type of node i.

4.4 Implementation of Objects Multi-Relationships Prediction

4.4.1 Link Prediction Model

The link prediction model can predict the relationships between the vehicles and the relationships between the pedestrians and the ego vehicle. An encoder-decoder architecture is constructed in link prediction model. In the graph $G = (V, E)$, with vertex set V and edge set E, the encoder maps each entity $v_i \in V$ to a real valued vector e_i. This process is known as embedding the entity into a vector space. The vector e_i captures the semantic information of the entity based on its relationships and context within the graph. Essentially, the encoder compresses the information about the entity into a lower-dimensional vector that is easier to work with computationally while retaining as much relevant information as possible. With the embeddings produced by the encoder, the decoder scores *(subject, relation, object)*-triples. The DistMult decoder implemented in this task scores the triple *(s, r, o)* with the function of

$$f(s, r, o) = e_s^T M_r e_o \tag{4.2}$$

which every relation r is associated with a diagonal matrix M_r. The score illustrates the possibility of the triplet – the higher the score is, the more possible is the triplet. The triplet with the highest scoring is the result of the edge.

There are two RGCN convolutional layers in the encoder and a DistMult decoder. During training, the loss function is binary cross-entropy loss with logits.

4.4.2 Link Prediction Model with Penalty Mechanism

In order to further enhance the performance of the RGCN+DistMult, the weight factor is considered because it can weigh the importance of different neighboring nodes. However, the self-attention mechanism has a drawback in its interpretability, and the result is not as expected. Thus, a penalty mechanism that regularizes the loss function during training is implemented. As the link prediction model tries to predict two kinds of edges which are divided into 5 classes:

Table 4.1: Link prediction edge types

Car-Car Relationship	‘nochange’ ‘movenear’ ‘moveaway’
Pedestrian-Ego Relationship	‘cross’ ‘notcross’

The results of the model illustrate that the model may confuse the two kinds of relationships, *e.g.,* mistaking the edge between the pedestrian and the ego vehicle with a relationship of ‘notcross’ for the car-car relationship ‘nochange’. In order to correct such a misjudgment, a judging algorithm is added to each epoch of training results – to determine whether the obtained relationship matches the two objects corresponding to the edge. If the nodes corresponding to the edge are cars, and the result of the training epoch is relationship ‘cross’ or ‘notcross’, then the prediction is negative; if a pedestrian is one of the nodes of the edge, and the output of the edge is ‘nochange’, ‘movenear’, or‘moveaway’, then the prediction is also negative. A penalty is added to the negative term in the loss function to emphasize the misprediction.

4.4.3 Rule Reasoning Model

The rule reasoning task is to predict the unknown edges in the graph based on the existing information. As in Figure 4.4, there are already edges between every two vehicles and edges between each pedestrian and the ego vehicle in the constructed multi-relational graphs. However, there are no existing edges between the pedestrians and the surrounding vehicles, which is the aim of the prediction.

The rule reasoning is a supervised task. Therefore, the unknown edges are labeled with a rule-based method as the ground truth. The rule is proposed in Table 4.2 and is based on common sense and general principles of traffic safety. The existing relationships have the same reference object which is the ego vehicle. When a vehicle is moving closer to the ego vehicle and a pedestrian is crossing the lane where the ego car is driving, which means the vehicle is getting closer to a crossing person, it has a high risk and is encoded as 2. When a vehicle is moving near the ego vehicle while the pedestrian is not crossing, or when a vehicle is not moving closer and the pedestrian is crossing, the risk should be lower but still present a moderate risk due to the unpredictable nature. This situation is labeled as Mid Risk and encoded as 1. If the vehicle is not coming closer and the pedestrian is not crossing, the risk is minimal. There's little to no interaction between the vehicle and the pedestrian that would cause concern. This situation is labeled as Low Risk and encoded as 0.

Table 4.2: Rule-based reasoning principle

Surrounding-Ego Vehicle Relationship	Pedestrian-Egocar Relationship	Rule-Reasoning Relationship
Movenear	Crossing	High Risk (2)
Movenear	NotCrossing	Mid Risk (1)
Nochange / Moveaway	Crossing	Mid Risk (1)
Nochange / Moveaway	NotCrossing	Low Risk (0)

According to Yang et al. (2015), the rule reasoning is based on the link prediction model. Within the relationship that satisfies the Horn rules, $B_1(a, b) \wedge B_2(b, c) \Rightarrow H(a, c)$, the scoring of the rule-reasoning relationship is $e_a^T(M_1M_2)e_c$, since the scoring function between a and b is $e_a^T M_1 e_b$, and the scoring function between b and c is $e_b^T M_2 e_c$. Thus, a link prediction model which is exactly the same as the model structure in 4.4.1, gains the embeddings e_a and e_c, and also the diagonal matrix M_1, M_2. Then, another rule reasoning model is constructed with a similar structure as the link prediction model. However, the gradients are disabled in the encoder of the link prediction model during the training. The decoder scores the triplet (e_a, r, e_c), and gets the diagonal matrix $M_{a,c}$. The loss function is calculated with MSE, and the ground truth is $M_1 * M_2$.

Chapter 5

Results and Analysis

In this chapter, the performance of various graph-based models on node classification and link prediction tasks is evaluated. For node classification, the RGCN model is utilized and compared with benchmark models to show its performance. The results are illustrated in accuracy and F1 Score. In the link prediction task, the RGCN+DistMult model is evaluated with accuracy and hits@2 scores. This chapter also discusses the effect of adding a penalty mechanism. Lastly, the results of the rule reasoning task are presented and discussed.

5.1 Node Classification Task

For node classification, the RGCNConv from PyG is utilized. The number of RGCNConv layers which means the depth of the model is 2 and the number of hidden channels is set to be 16. The training epoch is 100, the learning rate is 0.01 and the optimizer is Adam. The dataset is from *KITTI Object Detection Evaluation 2012 Left Color Images*. There are 480 multi-relational graphs generated from the image series as the whole dataset, and the train-test-split-ratio is $80 : 20$.

In this project, a standard GCN model is used as a benchmark to evaluate the performance of the RGCN model for node classification tasks. The GCN is a widely recognized model for processing graph-structured data, making it an appropriate baseline for comparison.

Meanwhile, a Graph Attention Network (GAT) model is used as another benchmark to evaluate the performance of the RGCN model. The GAT model introduces attention mechanisms to graph convolution, allowing it to weigh the importance of neighboring nodes differently.

Table 5.1: Parameters and factors of the RGCN model and the benchmarks

	RGCN model	GCN (Benchmark)	GAT (Benchmark)
Layer	RGCNConv	GCNConv	GATConv
Depth	2	2	2
Input Channel	2	2	2
Output Channel	2	2	2
Hidden Channel	16	16	16
Training Epoch	100	100	100
Learning Rate	0.01	0.01	0.01
Optimizer	Adam	Adam	Adam
Dataset	KITTI	KITTI	KITTI
Training set size	384 graphs	384 graphs	384 graphs
Test Set size	96 graphs	96 graphs	96 graphs

From Table 5.1, it is clear that all the parameters and factors are kept to be the same except for the convolutional layers used in each model. The experiment is repeated 10 times to reduce the error. Each time, the dataset is split randomly and the results of the experiment are shown as follows:

Table 5.2: Results of the models of the node classification task

	RGCN model	GCN (Benchmark)	GAT (Benchmark)
Accuracy	0.85	0.77	0.68
F1 Score	0.89	0.80	0.66

- RGCN has the highest accuracy and F1 Score among the three models, significantly outperforming both GCN and GAT. This suggests that RGCN is more effective at correctly predicting the outcomes in the given multi-relational driving scenario. It is also more balanced in terms of precision and recall, making it a more reliable model for this task.
- The GCN model also performs reasonably well but falls short compared to RGCN, indicating that while GCN can handle graph data, it struggles with the complexities introduced by multiple types of relationships between nodes.
- GAT has the lowest results in both accuracy and F1 Score, indicating that while attention mechanisms can be powerful, they may not be as effective in this particular context where understanding multiple types of relationships is crucial.

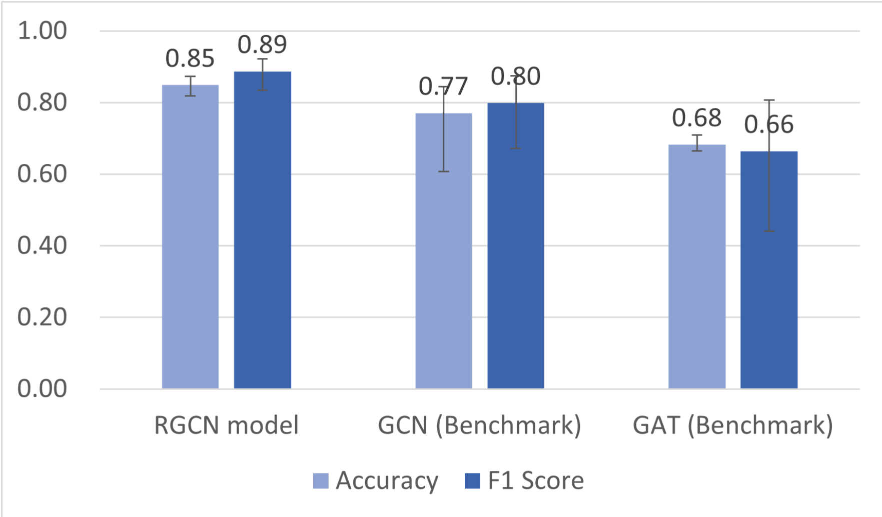


Figure 5.1: Visible result of the node classification task

As Figure 5.1 shows, the error bars in GCN and GAT are longer which also indicates that the benchmark models are less robust while predicting.

The results strongly suggest that RGCN is the most appropriate model for tasks involving multiple relationships from the driving scenarios and it indeed outperforms in the node classification task.

5.2 Link Prediction Task

In the link prediction task, the RGCNConv from PyG is also utilized as the convolutional layer. The number of RGCNConv layers which means the depth of the model is 2 and the number of hidden channels is set to be 8. The training epoch is 100, the learning rate is 0.01 and the optimizer is Adam. The dataset is from the PIE video clip SET_01. There are 315 multi-relational graphs generated from the video clips as the whole dataset, and the train-test-split-ratio is 80/20. The benchmark model is a vanilla DistMult Decoder without an encoder. It is used to evaluate the performance of the RGCN framework for link prediction tasks. Meanwhile, the RGCN+penalty is implemented by adding a penalty factor into the loss function, and the other parameters are set to be exactly the same as the RGCN+DistMult model. To evaluate the performance of the penalty factor, the RGAT+DistMult which utilized the Relational Graph Attention Network (RGAT) model is used as the benchmark in this task. The factors and parameters of these four models are shown in Table 5.3:

Table 5.3: Parameters and factors of the models used in link prediction experiments

	RGCN+DistMult model	DistMult (Benchmark)	RGCN+penalty model	RGAT+DistMult (Benchmark)
Layer	RGCNConv	-	RGCNConv	RGATConv
Depth	2	-	2	2
Number of Relations	5	5	5	5
Hidden Channel	8	8	8	8
Training Epoch	100	100	100	100
Learning Rate	0.001	0.001	0.001	0.001
Loss Function	Binary Cross Entropy Loss with Logits	Binary Cross Entropy Loss with Logits	Binary Cross Entropy Loss with Logits + penalty	Binary Cross Entropy Loss with Logits
Optimizer	Adam	Adam	Adam	Adam
Dataset	PIE	PIE	PIE	PIE
Training set number	252 graphs	252 graphs	252 graphs	252 graphs
Test Set number	63 graphs	63 graphs	63 graphs	63 graphs

The overall results of these four models is presented in Table 5.4 and analyzed as following:

- RGCN+DistMult achieved a solid overall accuracy of 0.82 and an excellent hits@2 score of 0.95. This indicates that the model is fairly reliable in predicting the correct relationships, with a high likelihood of including the correct relationship within the top 2 predictions.
- As the baseline, the basic DistMult model had a lower overall accuracy (0.74) and hits@2 (0.87). The RGCN+DistMult model outperforms DistMult in almost all metrics. This suggests that incorporating the

Table 5.4: Results of the models of the link prediction task

	Accuracy	Hits@2	Vehicle accuracy	Pedestrian accuracy	Vehicle hits@2	Pedestrian hits@2
RGCN + DistMult	0.82	0.95	0.80	0.91	0.94	0.99
DistMult benchmark	0.74	0.87	0.84	0.03	0.90	0.67
RGCN + penalty	0.86	0.95	0.86	0.84	0.94	0.98
RGAT + DistMult benchmark	0.61	0.84	0.63	0.49	0.86	0.74

RGCN architecture significantly improves the model's performance in understanding and predicting relationships.

- RGCN+Penalty achieved the highest overall accuracy (0.86) among all models, though its hits@2 (0.95) was slightly lower than the RGCN+DistMult. The penalty mechanism likely helped the model to focus on correct edge classifications, especially in more challenging cases.

- The performance of the RGAT+DistMult model which is the benchmark was the lowest, with an overall accuracy of 0.61 and hits@2 of 0.84. This indicates that the RGAT architecture may not be as effective as RGCN for this particular task.

When focusing on the comparison between the RGCN+DistMult and DistMult benchmark as in Figure 5.2, it is obvious that the overall performance is better when predicting the relationships between the dynamic objects in the driving scenarios. The overall improvement of the RGCN performance is about 10% compared to the baseline. The main improvement is in the aspect of predicting the relationships between the pedestrians and the ego-vehicle, which is illustrated in pedestrian accuracy and hits@2. When tracing the output of the prediction, the DistMult mistakenly categorizes many of what are actually pedestrian-ego relationships into car-car relationships. When it comes to the RGCN+DistMult model, the performance of distinguishing the pedestrian-ego relationships and the car-car relationships is much better. RGCN is much more sensitive to the multi-relational data. The reason why the RGCN model got a bit lower accuracy in predicting the car-car relationships might be due to the model focusing more on complex relational data.

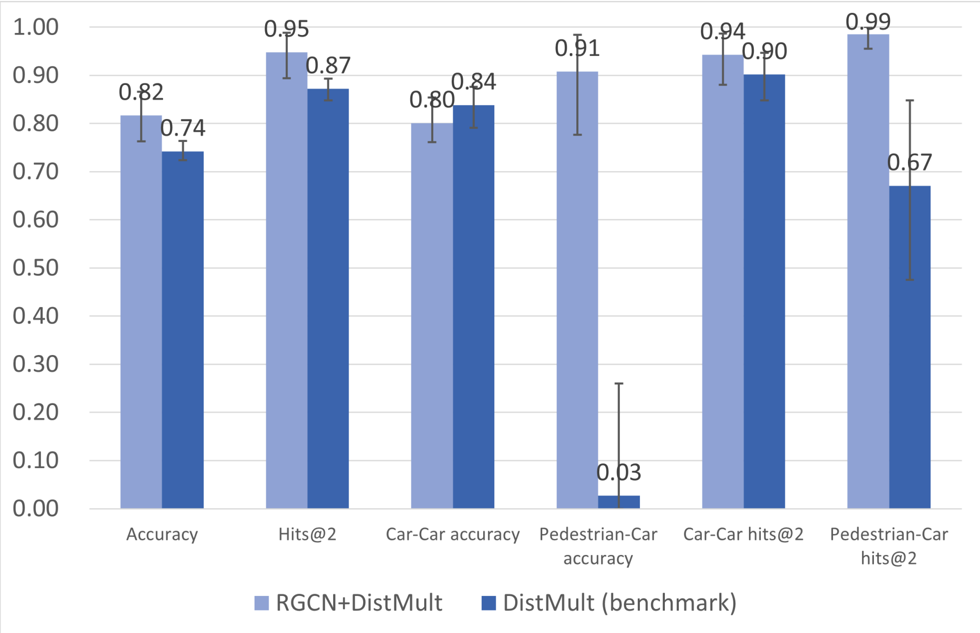


Figure 5.2: Comparison of results of the RGCN+DistMult and the DistMult benchmark

The RGCN+Penalty model significantly outperforms the RGAT across all metrics, indicating that the penalty mechanism introduced helps to regularize the model effectively, leading to better generalization and handling of relationships within the graph.

RGAT uses an attention mechanism to weigh the importance of different neighboring nodes. However, in this particular task, it seems that the attention mechanism used by RGAT may not be effective in capturing the relationships as the RGCN with an added penalty. The penalty is more targeted to distinguish the car-car relationships from the pedestrian-car relationships. This might help the model by constraining the learned parameters or relationships, thus reducing overfitting. The error bar in Figure 5.3 illustrates that the RGCN with penalty has higher robustness than the

RGAT one. This may also be due to the attention mechanism adding more weight parameters to the model, which can increase training time especially if the input data for the model are complex. When training with the same size of data and the same training epochs, the performance is not quite reliable. The results suggest that the penalty mechanism is more suitable for this particular situation. The penalty can also be highlighted for its clear structure and good interpretability.

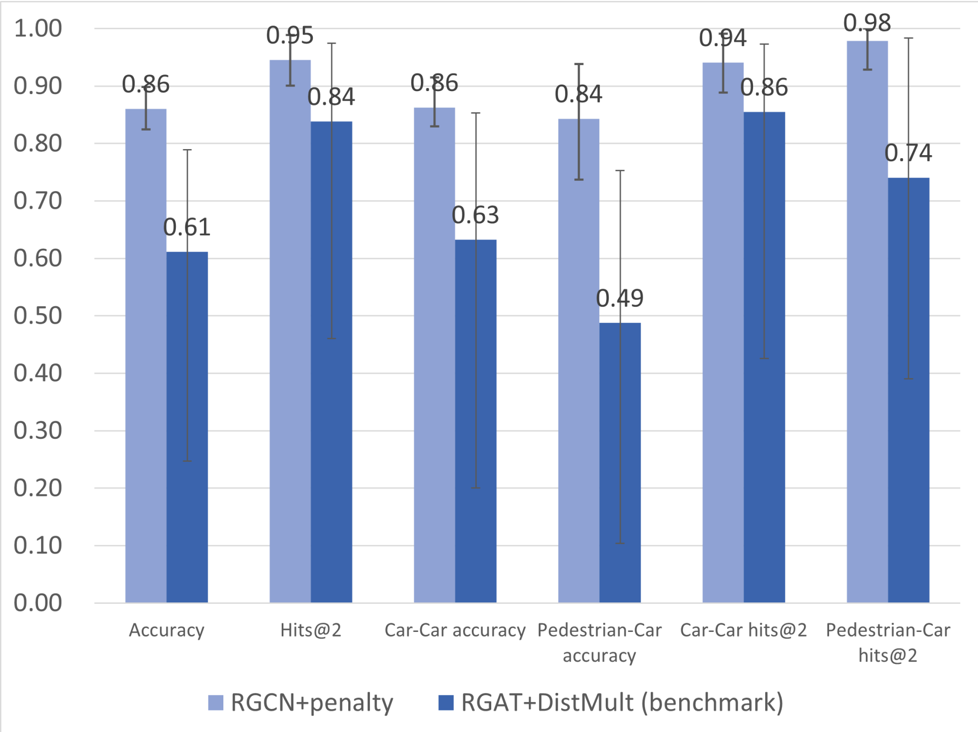


Figure 5.3: Comparison of results of the RGCN+penalty and the RGAT benchmark

On whether the penalty has improved the performance of the RGCN, the comparison is presented in Figure 5.4. The introduction of a penalty improves 5% of the overall accuracy and car-car-related metrics, but slightly reduces Hits@2 and pedestrian-related metrics. This suggests that while the penalty helps in improving the generalization and robustness of the model, it might also be restricting some of the flexibility that allows the base RGCN to achieve higher performance in pedestrian-related predictions. There may exist a trade-off because the penalty emphasizes the difference between the car-car relationships and the pedestrian-car relationships, it may ignore some of the inner differences (*e.g.,* $crossing$ or $notcrossing$ in the pedestrian-car relationships). This might explain the slight drop in pedestrian-related metrics.

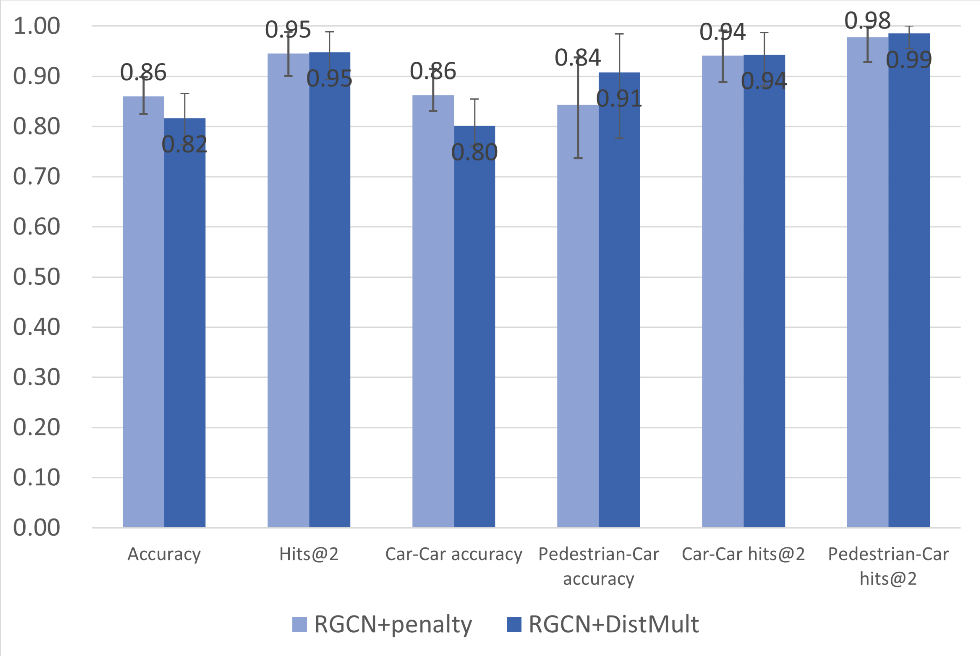


Figure 5.4: Comparison of results of the RGCN+penalty and the RGCN+DistMult

5.3 Rule Reasoning Task

Figure 5.5 displays two metrics related to predicting the risk level of the unknown edges between pedestrians and the surrounding vehicles using a rule-based reasoning model. As it is an extension of the link prediction task, the parameters of the model are quite similar to the RGCN+DistMult model.

Accuracy measures the proportion of correct predictions made by the model out of all predictions. In this context, it indicates that the model correctly identified the risk relationship between nodes about 0.78 of the time. The Hits@2 value is very high at 0.91 probably due to there being only 3 classes of relationships. This suggests a reasonable level of performance, but there is room for improvement and it is not reliable enough for real-world risk assessment.

The model performs better in terms of Hits@2 than pure accuracy which can be taken as the Hits@1. This indicates that while the model might not always hit the exact prediction, it is very good at narrowing down the possible correct relationships.

The discrepancy between accuracy and Hits@2 suggests that the model could benefit from further refinement to improve its precision, but its current form already offers a robust method for identifying potential risk relationships between nodes.

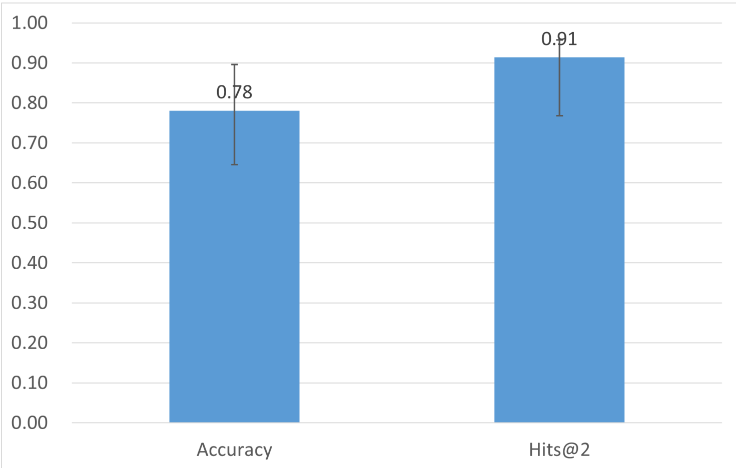


Figure 5.5: Results of the rule-reasoning task

Chapter 6

Conclusions and Future work

This chapter presents the overall conclusions drawn from the research and explores the potential directions for future work. The conclusions summarize the effectiveness of using GNNs, particularly the RGCNs, in understanding dynamic driving scenarios and predicting multi-relational graph structures. The chapter also discusses the limitations of the current approach, identifying areas where the model could be improved for better precision and applicability in real-world scenarios. Finally, it outlines several promising avenues for future research.

6.1 Conclusions

This project explored the use of GNNs, particularly the RGCNs, for predicting multi-relational graph structures involving interactions between vehicles and pedestrians of dynamic driving scenarios. The main purpose is to build a model that only relies on the pixel-level unstructured data from the monocular camera so that the context of the traffic can be understood and the relationships can be predicted. A conversion process is implemented to transform the unstructured data into graph-structured data which is spatial-temporal and multi-relational. This process involves tracking objects and comparing time-series data. Then the RGCNs models take this graph structure data to predict the behavior and the relationships of the objects. The prediction is divided into several tasks: node classification, link prediction, and rule reasoning. The node classification RGCN model can predict whether the car is in the same direction as the ego-vehicle or not. As a comparison, a basic GCN model and a GAT model are adopted as benchmarks to demonstrate the effect of the performance of the RGCN. The RGCN reached a high accuracy of 0.85. The link prediction task is to predict the car-car relationships and the pedestrian-ego relationships, which involves 5 categories of relationships. The model is composed of an RGCN encoder and a DistMult decoder. To evaluate the performance of the RGCN+DistMult, a DistMult model is implemented. The accuracy result of the RGCN+DistMult is 0.82 which is a 10% improvement from the DistMult model. Apart from that, the RGCN model with a penalty mechanism is also proposed aiming to enhance the performance of the link prediction and reach an accuracy of 0.86. The rule reasoning task is based on the link prediction task by using a rule-based method on the edges and classifying risk into three bands. The model ended up with a 0.78 accuracy.

In the end, the research question proposed in Chapter 1 can be reviewed:

- How to model the graph-based data in the context of a dynamic driving scenario?

  Section 4.2 proposed the detailed process of converting the unstructured data into graph-based data. The process includes object tracking, coordinates transformation, generating node and edge labels, and constructing the spatial-temporal multi-relational graphs. The framework remains sufficient information for subsequent prediction tasks.

- How much improvement can be achieved in predicting the vehicles' on-road behavior utilizing the RGCN model over the GCN models in the dynamic driving scenarios understanding?

  Section 4.3, 4.4.1 proposed the implementation of the RGCN in node classification and link prediction tasks. The results presented in section 5.1, 5.2 illustrate that compared to the benchmark model, the RGCN model improves the prediction accuracy by about 10%.

- Given the known relationships within the scenarios, can RGCN+DistMult model be used in reasoning the relationship of unknown edges?

  Section 4.4.3 proposes an approach based on RGCN+DistMult to reason the unknown edges from the existing relationships. The unknown edges illustrate the relationship of risk. The results of the rule reasoning task are presented in 5.3, and reach an overall accuracy of 0.78 which suggests that the RGCN+DistMult model can be used in reasoning the relationship of unknown edges.

To summarize, this project proposes a framework to understand the dynamic driving scenarios based on the RGCN. The proposed model demonstrates promising results when processing multi-relational data, effectively capturing the interactions between vehicles and pedestrians, which indicates that the RGCN-based approach has the potential for further development and application in advanced automated driving systems.

6.2 Limitations

Despite the promising results, several limitations need to be addressed to improve the model's precision and applicability in real-world scenarios.

- **Coordinate Transformation:** The vehicle's position coordinates are determined based on the bounding box center. This operation can lead to inaccuracies, especially when vehicles are at the edges of the camera's field of view. When the objects are at the edges, the objects in the camera's field of view may just be incomplete, and if the centroid of the bounding box continues to be used, then the coordinates are likely not to be the true centroid of the object and will be shifted. This may affect the judgment of the relationships. Additionally, when transforming to BEV, the OpenCV matrix is used instead of the camera caliberation parameters, which may also cause some inaccuracy.
- **Graph Structure and Features:** The graph structure used in this study is relatively simple, with limited features and a basic set of relationships between participants. The feature does not account for other critical factors like vehicle speed, acceleration, and trajectory, which are essential for understanding dynamic interactions in traffic scenarios. The graph structures are simple because they only contain the objects of vehicles and pedestrians, without considering objects like cyclists or motorcyclists, and static objects.
- **Dataset Limitations:** The datasets used for training and testing primarily consist of idealized scenarios with good lighting and standard road conditions. This limitation means that the model may not perform as well in challenging real-world conditions, such as poor lighting, extreme weather, or complex road environments. Future work should focus on testing the model in a broader range of scenarios to ensure robustness across different conditions.

6.3 Future work

To address these limitations and further improve the model, future work should focus on the following areas:

- Implementing camera calibration techniques to determine vehicle positions and account for perspective distortion more accurately.
- Expanding the classification approach to include dynamic features like speed, direction, and acceleration, could provide a deeper understanding of the interactions between participants.
- Designing more complex graph structures that include a broader range of participants and richer sets of relationships and features, could lead to more accurate predictions. Also enlarging the size of the dataset.
- Collecting and utilizing data from a wider variety of conditions, including challenging lighting, weather, and road scenarios, to ensure that the model can generalize well to all real-world situations.
- Integrating with hardware systems to enable real-time adjustments and control, making it feasible to conduct practical experiments in real-world scenarios. This would not only validate the model's performance in controlled environments but also provide insights into its effectiveness and reliability under diverse and dynamic conditions. Such integration could lead to the development of more robust and adaptable automated driving technologies.

References

Bharilya, V. and Kumar, N. (2023). Machine Learning for Autonomous Vehicle's Trajectory Prediction: A comprehensive survey, Challenges, and Future Research Directions. arXiv:2307.07527 [cs]. [Page 2.]

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc. [Page 12.]

Bordes, A., Weston, J., Collobert, R., and Bengio, Y. (2011). Learning structured embeddings of knowledge bases. *Proceedings of the AAAI Conference on Artificial Intelligence*, 25(1):301–306. [Page 12.]

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. (2014). Spectral networks and locally connected networks on graphs. [Page 10.]

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. [Pages ix and 19.]

Glorot, X., Bordes, A., Weston, J., and Bengio, Y. (2013). A semantic matching energy function for learning with multi-relational data. [Page 12.]

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. [Page 10.]

Li, Z., Zhao, Y., Zhang, Y., and Zhang, Z. (2022). Multi-relational graph attention networks for knowledge graph completion. *Knowledge-Based Systems*, 251:109262. [Page 9.]

Liu, Z. and Zhou, J. (2022). *Introduction to graph neural networks*. Springer Nature. [Page 10.]

Mylavarapu, S., Sandhu, M., Vijayan, P., Krishna, K. M., Ravindran, B., and Namboodiri, A. (2020a). Towards accurate vehicle behaviour classification with multi-relational graph convolutional networks. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 321–327. IEEE. [Pages 3, 9, and 15.]

Mylavarapu, S., Sandhu, M., Vijayan, P., Krishna, K. M., Ravindran, B., and Namboodiri, A. (2020b). Understanding dynamic scenes using graph convolution networks. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8279–8286. IEEE. [Pages 3, 9, and 15.]

Rasouli, A., Kotseruba, I., Kunic, T., and Tsotsos, J. K. (2019). Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In *International Conference on Computer Vision (ICCV)*. [Page 20.]

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. [Page 7.]

Scarselli, F., Tsoi, A. C., Gori, M., and Hagenbuchner, M. (2004a). Graphical-Based Learning Environments for Pattern Recognition. In Fred, A., Caelli, T. M., Duin, R. P. W., Campilho, A. C., and de Ridder, D., editors, *Structural, Syntactic, and Statistical Pattern Recognition*, pages 42–56, Berlin, Heidelberg. Springer Berlin Heidelberg. [Page 2.]

Scarselli, F., Tsoi, A. C., Gori, M., and Hagenbuchner, M. (2004b). Graphical-Based Learning Environments for Pattern Recognition. In Fred, A., Caelli, T. M., Duin, R. P. W., Campilho, A. C., and de Ridder, D., editors, *Structural, Syntactic, and Statistical Pattern Recognition*, Lecture Notes in Computer Science, pages 42–56, Berlin, Heidelberg. Springer. [Page 9.]

Schlichtkrull, M., Kipf, T. N., Bloem, P., Berg, R. v. d., Titov, I., and Welling, M. (2017). Modeling Relational Data with Graph Convolutional Networks. arXiv:1703.06103 [cs, stat]. [Pages ix, 10, 11, 12, and 22.]

Socher, R., Chen, D., Manning, C. D., and Ng, A. (2013). Reasoning with neural tensor networks for knowledge base completion. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc. [Page 12.]

Su, P. and Chen, D. (2022). Using fault injection for the training of functions to detect soft errors of dnns in automotive vehicles. In Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., and Kacprzyk, J., editors, *New Advances in Dependability of Networks and Systems*, pages 308–318, Cham. Springer International Publishing. [Page 14.]

Su, P. and Chen, D. (2024). Adopting graph neural networks to analyze human–object interactions for inferring activities of daily living. *Sensors*, 24(8):2567. [Page 15.]

Su, P., Kang, S., Tahmasebi, K. N., and Chen, D. (2023a). Enhancing Safety Assurance for Automated Driving Systems by Supporting Operation Simulation and Data Analysis. In *Proceeding of the 33rd European Safety and Reliability Conference*, pages 249–256. Research Publishing Services. [Page 14.]

Su, P., Wang, Y., Xiang, C., Wendel, E., Mishra, M., and Chen, D. (2024). Enhanced prognostics and health management in automated driving systems: Using graph neural networks to recognize operational contexts. In *2024 Prognostics and System Health Management Conference (PHM)*, pages 415–421. [Page 15.]

Su, P., Warg, F., and Chen, D. (2023b). A simulation-aided approach to safety analysis of learning-enabled components in automated driving systems. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 6152–6157. [Page 14.]

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2021). A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24. [Pages 10 and 15.]

Xu, Y., Wang, L., Wang, Y., and Fu, Y. (2022). Adaptive trajectory prediction via transferable gnn. [Page 15.]

Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. (2015). Embedding Entities and Relations for Learning and Inference in Knowledge Bases. arXiv:1412.6575 [cs]. [Pages 3, 12, 13, 22, and 34.]

Yu, S.-Y., Malawade, A. V., Muthirayan, D., Khargonekar, P. P., and Faruque, M. A. A. (2020). Scene-graph augmented data-driven risk assessment of autonomous vehicle decisions. [Page 28.]

Yu, S.-Y., Malawade, A. V., Muthirayan, D., Khargonekar, P. P., and Faruque, M. A. A. (2022). Scene-Graph Augmented Data-Driven Risk Assessment of Autonomous Vehicle Decisions. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7941–7951. Conference Name: IEEE Transactions on Intelligent Transportation Systems. [Pages 3 and 15.]

TRITA-ITM-EX 2024:595
Stockholm, Sweden 2025

www.kth.se